University of Peloponnese

Faculty of Economy, Management and Informatics
Department of Informatics and Telecommunications

Framework for the effective implementation of alternative teaching methods for Informatics

by Theodoropoulos Anastasios

A thesis submitted for the degree of
Doctor of Philosophy

Supervisor: Lepouras George

Tripolis 2017

Defense Committee

George Lepouras, Professor, Department of Informatics and Telecommunications, University of Peloponnese (Supervisor)

Costas Vassilakis, Professor, Department of Informatics and Telecommunications, University of Peloponnese (Member of steering committee)

Manolis Wallace, Assistant Professor, Department of Informatics and Telecommunications, University of Peloponnese (Member of steering committee)

Athanassios Jimoyiannis, Professor of Science and ICT in Education, Department of Social and Educational Policy, University of Peloponnese

Konstantinos Dimopoulos, Professor of Learning Materials, Department of Social and Educational Policy, Faculty of Social Sciences, University of Peloponnese

Dimitris Charitos, Assistant Professor, Faculty of Communication and Media Studies, University of Athens

Kostas Karpouzis, Associate Researcher at the Institute of Communication and Computer Systems (ICCS) of the National Technical University of Athens (NTUA)

This page was intentionally left mostly blank.

Abstract

The dynamic development of the CS field introduces new educational and pedagogical challenges, including the instructional design of teaching and learning. How can we teach our students better in such a growing and demanding field? Moreover, how can we motivate them and together have better learning results? These are the two key questions addressed in this dissertation.

There are several teaching methods used by educators and researchers in order to deal with the complexity and the needs of various cases. Alternative Teaching Methods (ATM) that support students' active involvement in the learning process, according to the constructivism and constructionism principles, could be very helpful. This thesis presents a student-centered framework for effective encompassing ATM in Computer Science Education (CSE). A framework for teaching can provide shared understandings, which can help improve the quality of instructional design, course and lesson planning, learning and assessment.

For the conception of the framework, I describe and analyze every known method, practice, strategy and teaching approach in CSE. Through an extensive analysis, I summarize the benefits of ATM and consider known issues of traditional teaching. In addition, I provide empirical evidence that alternative teaching provides increased motivation for learning and better learning outcomes. The empirical evidence comes from five research studies, with various ATM researched at different levels of education in Greek settings. Firstly, I conducted an experiment using Peer Learning and Collaboration techniques comparing them to traditional teaching, to secondary education students. Secondly, I researched Social Networks for assisted learning, through an observational study with undergraduate students of a Greek university. Next, I examined the Game Based Learning method, in order to teach basic programming concepts to primary school pupils. The fourth research study I conducted was about the use of Educational Robotics as alternate to teach programming and engineering concepts through a national competition. Finally, I made an experiment with the use of serious games that promote programming learning with a focus on personal learning characteristics. Both previous work and the empirical research of this thesis helped me shape the framework for alternative teaching (ATMF).

This thesis provides a number of contributions: Firstly, it presents a high level description/review of alternative teaching methods that are used in CSE, identifying strengths and areas for improvement. Next, it offers empirical evidence that alternative teaching methods in Greek education provide extra motivation for learning and better learning outcomes. Finally, it provides a conceptual framework based both on previous theoretical frameworks and models as well as empirical guidelines, for the effective use of alternative teaching methods in CSE. The framework does not teach computer science; rather, it focuses on computer science teaching. It is also not limited to the teaching of a specific computer science curriculum, neither is it limited to the teaching of a specific programming language nor to a specific programming paradigm and can be adapted to the teaching of any CS topic in any context and any level, from elementary school through high school to the university level.

Finally, the generality, limitations and potential of the proposed framework and empirical studies are discussed, and other factors affecting the learner's performance and motivation, such as personality traits, are briefly considered. Future directions building on the work described herein are presented and discussed.

Εκτεταμένη Περίληψη

Η δυναμική εξέλιξη του πεδίου της Πληροφορικής, εισάγει νέες εκπαιδευτικές και παιδαγωγικές προκλήσεις, συμπεριλαμβανομένου του σχεδιασμού και της μεθοδολογικής οργάνωσης της διδασκαλίας και της μάθησης. Πώς μπορούμε να διδάξουμε τους μαθητές μας καλύτερα σε ένα πεδίο όπως της Πληροφορικής, το οποίο αναπτύσσεται ταχύτατα και με ιδιαίτερες απαιτήσεις; Επιπλέον, πώς μπορούμε να τους δώσουμε κίνητρα επιτυγχάνοντας ταυτόχρονα καλύτερα μαθησιακά αποτελέσματα; Αυτά είναι τα δύο κύρια ερωτήματα που καλείται να απαντήσει η παρούσα διδακτορική διατριβή.

Οι εκπαιδευτικοί και οι ερευνητές διαθέτουν πληθώρα μεθόδων διδασκαλίας που μπορούν να εφαρμοστούν και οι οποίες στοχεύουν στη μείωση της πολυπλοκότητας και στην υποστήριξη της εκπαίδευσης σε διαφορετικές καταστάσεις στην Εκπαίδευση της Πληροφορικής. Δυστυχώς, αυτές εστιάζουν σε διαφορετικά θέματα και επίπεδα της εκπαιδευτικής διαδικασίας και μεμονωμένα αδυνατούν να παράσχουν ολοκληρωμένες απαντήσεις στα βασικά ερευνητικά μας ερωτήματα.

Μια συχνή κατηγοριοποίηση των μεθόδων διδασκαλίας είναι σε παραδοσιακές και σε εναλλακτικές. Κύριο χαρακτηριστικό των εναλλακτικών μεθόδων διδασκαλίας (ΕΜΔ) είναι η τοποθέτηση του μαθητή στο κέντρο της εκπαιδευτικής διαδικασίας. Προς αυτήν την κατεύθυνση, η παρούσα διατριβή παρουσιάζει ένα μαθητο-κεντρικό πλαίσιο για την αποτελεσματική εφαρμογή ΕΜΔ στην Εκπαίδευση της Πληροφορικής. Ένα πλαίσιο διδασκαλίας που μπορεί να βελτιώσει την ποιότητα του διδακτικού σχεδιασμού, του πλάνου μαθήματος, της μάθησης αλλά και της αξιολόγησης.

Για την δημιουργία του Πλαισίου Εναλλακτικών Μεθόδων Διδασκαλίας (ΠΕΜΔ), αρχικά περιγράφω και αναλύω κάθε γνωστή μέθοδο, πρακτική, στρατηγική και προσέγγιση διδασκαλίας στην Εκπαίδευση της Πληροφορικής. Μέσω μιας εκτενούς βιβλιογραφικής ανάλυσης παρουσιάζω τα κύρια πλεονεκτήματα των ΕΜΔ και τα προβλήματα που έχει η παραδοσιακή διδασκαλία. Επιπλέον, παρέχω εμπειρικά στοιχεία που στηρίζουν ότι η εναλλακτική διδασκαλία παρέχει αυξημένα κίνητρα για μάθηση και καλύτερα μαθησιακά αποτελέσματα. Τα εμπειρικά στοιχεία προέρχονται μέσα από πέντε ερευνητικές μελέτες, με διάφορες ΕΜΔ σε διάφορα επίπεδα της Ελληνικής εκπαίδευσης.

Στην πρώτη ερευνητική μελέτη, διεξήγαγα ένα πείραμα με τη χρήση μεθόδων Ομότιμης και Συνεργατικής μάθησης σε σχέση με την παραδοσιακή διδασκαλία, σε μαθητές δευτεροβάθμιας εκπαίδευσης. Στη μελέτη αυτή αξιολόγησα την απόδοση των μαθητών και εξέτασα τις στάσεις τους απέναντι στην εναλλακτική διδασκαλία. Τα αποτελέσματα έδειξαν καλύτερες επιδόσεις στις ομάδες μαθητών που χρησιμοποιήθηκαν ΕΜΔ, καθώς και ιδιαίτερα θετικές στάσεις. Το συγκεκριμένο πείραμα, χρησιμοποιήθηκε ως είσοδος κατά τη διάρκεια του σχεδιασμού του ΠΕΜΔ.

Η δεύτερη ερευνητική μελέτη, αποτέλεσε επίσης είσοδο στο προτεινόμενο ΠΕΜΔ. Πρόκειται για μια μελέτη παρατήρησης όπου διερεύνησα την αξιοποίηση γνωστού Κοινωνικού Δικτύου, ως υποστηρικτικό εργαλείο στην τριτοβάθμια εκπαίδευση. Μέσα από ποιοτικά και ποσοτικά δεδομένα εξέτασα τρόπους με τους οποίους τα Κοινωνικά Δίκτυα μπορούν να αξιοποιηθούν αποτελεσματικά στη διδασκαλία και τη μάθηση.

Στην επόμενη ερευνητική μελέτη, εξέτασα ως μέθοδο διδασκαλίας τα Ψηφιακά Παιχνίδια με σκοπό την εκμάθηση βασικών αρχών προγραμματισμού σε μαθητές δημοτικών σχολείων. Οι μαθητές μέσα από μια ολιγόωρη επίσκεψή τους στο τμήμα Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Πελοποννήσου, προγραμμάτισαν σε ζευγάρια (Pair Programming). Σε αυτήν τη μελέτη διερεύνησα παράγοντες όπως η ικανοποίηση, η αξιολόγηση της γνώσης και η προθυμία χρήσης παιχνιδιών στο μέλλον από τους μαθητές. Επιπλέον, εξέτασα εάν μια σύντομη δραστηριότητα όπως αυτή μπορεί να οδηγήσει σε μακροχρόνια παρακίνηση των μαθητών. Τα αποτελέσματα έδειξαν ότι οι μαθητές απόλαυσαν τον συγκεκριμένο τρόπο διδασκαλίας και ότι θεωρούν πως έμαθαν βασικές αρχές προγραμματισμού. Επίσης προτιμούν τον προγραμματισμό σε ζεύγη και εκτιμούν ότι μπορούν να μάθουν καλύτερα βοηθώντας ο ένας τον άλλον. Τέλος, φαίνεται ότι μια επίσκεψη λίγων ωρών δεν ήταν αρκετή για να παρακινήσει τους μαθητές σε μακροπρόθεσμα αποτελέσματα. Η ερευνητική μελέτη αυτή πραγματοποιήθηκε μετά το σχεδιασμό του ΠΕΜΔ, αποτελώντας μια πρώτη αξιολόγηση του, μέσα από την οποία κάποια στοιχεία του ενισχύθηκαν και επιπλέον κάποια νέα προστέθηκαν.

Η τέταρτη ερευνητική εμπειρική μελέτη που διεξήγαγα, σχετίζεται με την αξιοποίηση της Εκπαιδευτικής Ρομποτικής για τη διδασκαλία αρχών προγραμματισμού αλλά και μηχανικής. Για το σκοπό αυτό, κατέγραψα ποιοτικά δεδομένα με τις στάσεις των εκπαιδευτικών σχετικά με τη συμμετοχή των μαθητών τους στον Πανελλήνιο διαγωνισμό Εκπαιδευτικής Ρομποτικής και διερεύνησα την ανάπτυξη δεξιοτήτων, κινήτρων αλλά και πιθανά μαθησιακά οφέλη. Τα αποτελέσματα έδειξαν πολύ υψηλό επίπεδο εμπλοκής και κινητοποίησης τόσο καθηγητών όσο και μαθητών. Με την ρομποτική ως ΕΜΔ αναπτύχθηκαν στους μαθητές ικανότητες όπως συνεργασία, επίλυση προβλημάτων και δημιουργικότητα καθώς επίσης και ανάπτυξη υπολογιστικής σκέψης. Και αυτή η ερευνητική μελέτη πραγματοποιήθηκε μετά το σχεδιασμό του ΠΕΜΔ, αποτελώντας ένα δεύτερο στάδιο αξιολόγησής του.

Στην συνέχεια, πραγματοποίησα ένα τελικό πείραμα με τη χρήση σοβαρών παιχνιδιών που προάγουν την εκμάθηση προγραμματισμού, σε μαθητές δευτεροβάθμιας εκπαίδευσης και με επίκεντρο τα προσωπικά μαθησιακά χαρακτηριστικά. Αρχικά διερεύνησα τις στάσεις των μαθητών από τις δραστηριότητες των παιχνιδιών με σκοπό να βρεθεί η ποιότητα της μαθησιακής τους εμπειρίας. Στη συνέχεια, οι στάσεις αυτές συσχετίστηκαν με το γνωστικό τους προφίλ και επιπλέον, αξιολογήθηκε η απόδοσή τους από τα παιχνίδια η οποία και συσχετίσθηκε επίσης με το γνωστικό τους στυλ. Τα αποτελέσματα έδειξαν ότι τα παιχνίδια που αξιοποιήθηκαν μπορούν να αποτελέσουν κατάλληλα περιβάλλοντα μάθησης μέσα στα σχολεία, καθώς παρέχουν μια υψηλής ποιότητας μαθησιακή εμπειρία. Επίσης, το γνωστικό στυλ διαπιστώθηκε ότι είναι ένα σημαντικό μαθησιακό χαρακτηριστικό που θα πρέπει να λαμβάνεται υπόψη κατά τη χρήση ψηφιακών παιχνιδιών για την εκμάθηση προγραμματισμού. Ως εκ τούτου, η συγκεκριμένη ερευνητική μελέτη αποτέλεσε την τρίτη και τελευταία αξιολόγηση του προτεινόμενου πλαισίου, και όπως και οι προηγούμενες ενίσχυσε κάποια στοιχεία και αφού προσέθεσε κάποια ακόμα, οριστικοποίησε το ΠΕΜΔ.

Συνοψίζοντας, τόσο προηγούμενη σχετική έρευνα, όσο και οι πέντε εμπειρικές μελέτες αυτής της διατριβής αποτέλεσαν τις εισόδους για την δημιουργία και τελική διαμόρφωση του ΠΕΜΔ στην Εκπαίδευση της Πληροφορικής. Οι συνεισφορές της παρούσας διδακτορικής διατριβής είναι οι εξής: 1) παρουσιάζει ένα υψηλό επίπεδο βιβλιογραφικής επισκόπησης όλων των γνωστών ΕΜΔ που χρησιμοποιούνται στην Πληροφορική, προσδιορίζοντας τα δυνατά σημεία τους, 2) παρέχει εμπειρικά στοιχεία ότι οι ΕΜΔ στην Ελληνική εκπαίδευση δίνουν επιπλέον κίνητρα για μάθηση αλλά και καλύτερα μαθησιακά αποτελέσματα, και 3) παρουσιάζει ένα εννοιολογικό πλαίσιο που βασίζεται τόσο σε προηγούμενα θεωρητικά πλαίσια και μοντέλα όσο και σε εμπειρικές κατευθυντήριες γραμμές για την αποτελεσματική χρήση των ΕΜΔ στην Εκπαίδευση της Πληροφορικής. Το πλαίσιο εστιάζει στη διδακτική της Πληροφορικής χωρίς να περιορίζεται στη διδασκαλία ενός συγκεκριμένου προγράμματος σπουδών, κάποιας συγκεκριμένης γλώσσας προγραμματισμού ή κάποιου συγκεκριμένου μαθήματος.

Τέλος, στην παρούσα διδακτορική διατριβή συζητούνται η γενίκευση, οι περιορισμοί και οι δυνατότητες του προτεινόμενου πλαισίου καθώς και των επιμέρους εμπειρικών μελετών. Επίσης, εξετάζονται σύντομα πρόσθετοι παράγοντες οι οποίοι φαίνεται να επηρεάζουν την απόδοση και τα κίνητρα των μαθητευομένων, όπως τα χαρακτηριστικά της προσωπικότητας. Κλείνοντας, παρουσιάζονται μελλοντικά σχέδια και πιθανές προεκτάσεις που βασίζονται στην παρούσα διδακτορική διατριβή.

Acknowledgements

At this point I have the opportunity to thank all the people, that without their continuous support, it wouldn't be possible to finish this thesis. This is obviously the only section of this dissertation that I write without having it reviewed and I am so glad about it.

Firstly, I want to express my gratitude to my supervisor George for all the guidance he provided me. George, thank for your supervision, your frank thoughts and tireless support. You gave me a tremendous amount of freedom to explore what is fundamentally a muddy area while continually encouraging me to find ways to objectively measure success.

My advising committee members Manolis and Costas provided valuable feedback that guided both my research and the presentation of it. It has been a pleasure working with them. Costas your insight, patience, expertise, and concern for all matters of the university is admirable, and I could not imagine a better mentor. Manolis, thanks for your intellectual contributions and your unwavering support of my research, and for nurturing my growth as an academic. Despite the many challenges of this past year, you have made time at all the critical stages to ensure I stayed true to my goals, and for that I will always be grateful.

However, this thesis would not be possible without Angela. Angela assisted with my research studies based on her experience in education and psychology. She is an intelligent, motivated, and sociable young woman and the impact of learning in her life continually brought this thesis close to home. Angela, I owe you a big thank for your understanding, support, encouragement, and love since the beginning. What a privilege it has been to be advised and guided by you during every step of this journey.

I am also grateful for the love and encouragement from my parents. You taught me, by example, the irreplaceable value of hard work. Thank you for all the multi-level support.

I also want to thank my friends Manolis and Asbjorn for proofreading parts of this thesis.

Last, but definitely not least Dimitra for enduring my complaining, for being my supporter, and for watching our kids while I was writing. Earning a Ph.D. with a family growing during that journey, takes a team effort on the home front. Dimitra thank you for helping me keep life's priorities straight. I could not have done it without you! Your smiling eyes help put life back in perspective.

Throughout my career in education, many colleagues have enriched my thinking about learning processes and applications. My learning has been broadened by many outstanding students, teachers, counselors, administrators, with whom I have worked. Sincere thanks go to all the student collaborators for their assistance on research studies of this thesis. Although I serve as CS teacher, the students that participated in my studies helped me to understand the space of inspiring teaching and learning.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Anastasios D. Theodoropoulos)

to Dimitra, Aris and Dimitris

Table of Contents

DEFENSE COMMITTEE .... **II**

ABSTRACT .... **IV**

ΕΚΤΕΤΑΜΕΝΗ ΠΕΡΙΛΗΨΗ .... **V**

ACKNOWLEDGEMENTS .... **VIII**

DECLARATION .... **IX**

LIST OF FIGURES .... **XV**

LIST OF TABLES .... **XVII**

PART I - INTRODUCTION **1**

1 INTRODUCTION **2**

1.1 RESEARCH MOTIVATION 2
1.2 RESEARCH GOAL AND QUESTIONS 4
1.3 DEFINITIONS 5
1.4 THESIS OUTLINE 5
1.5 SUMMARY 7

PART II - MAPPING THE TERRITORY **8**

2 THE DISCIPLINE **9**

2.1 DEFINING THE TERMINOLOGY 9
2.2 SKILLS AND CHARACTERISTICS 11
2.3 TAXONOMY FOR CSE 12
2.4 SUMMARY 14

3 TEACHING AND LEARNING **15**

3.1 THE TEACHING 15
3.2 THE LEARNING 17
3.3 TRADITIONAL STRATEGIES 21
3.4 ALTERNATIVE METHODS 23

3.4.1 Peer Learning 23
3.4.2 Problem Based Learning 25
3.4.3 Project-Based Learning 27
3.4.4 Studio-Based Learning 28
3.4.5 Inquiry-Based Learning 30
3.4.6 Process Oriented Guided Inquiry Learning 31
3.4.7 Team Learning 32
3.4.8 Game-Based Learning 35
3.4.9 Educational Robotics 38
3.4.10 Non-textual programming 40
3.4.11 Contextualized Learning 43
3.4.12 CS Unplugged 45
3.4.13 Subgoal Learning 46
3.4.14 Parson's Programming Puzzles 48

3.4.15 Extreme Programming ... *49*
3.4.16 Program Visualization ... *50*
3.4.17 Competency-Based Learning ... *51*
3.4.18 Social Networks for Assisted Learning ... *54*
3.4.19 Emerging Technologies ... *55*
3.5 SUMMARY ... 57

PART III - THE FRAMEWORK ... 58

4 FRAMEWORK FOR ALTERNATIVE TEACHING ... 59
4.1 A NEW CONCEPTUAL FRAMEWORK ... 59
4.2 THE FRAMEWORK' S PRINCIPLES ... 60
4.3 RESEARCHING - OTHER FRAMEWORKS AND MODELS ... 62
4.3.1 The Model of Educational Reconstruction ... *62*
4.3.2 The Framework for Teaching ... *64*
4.3.3 First Principles of Instruction ... *66*
4.3.4 The K-12 CS Framework ... *67*
4.3.5 Taxonomies ... *70*
4.3.6 Learning Technology System Architecture ... *73*
4.4 ANALYSIS - CONCEPTING ... 74
4.5 SHAPING THE FRAMEWORK ... 80
4.6 SUMMARY ... 81

5 RESEARCH STUDY 1: PEER LEARNING ... 82
5.1 INTRODUCTION ... 82
5.1.1 CS in the Greek Secondary Education ... *84*
5.2 METHOD ... 85
5.2.1 First Experimental Condition ... *87*
5.2.2 Second Experimental Condition ... *89*
5.2.3 Third Experimental Condition ... *90*
5.3 RESULTS ... 90
5.4 DISCUSSION ... 98
5.5 CONCLUSIONS ... 100
5.6 ELEMENTS IN THE ATMF ... 100
5.7 SUMMARY ... 102

6 RESEARCH STUDY 2: SOCIAL NETWORKS IN EDUCATION ... 103
6.1 INTRODUCTION ... 103
6.2 METHOD ... 105
6.3 RESULTS ... 107
6.4 DISCUSSION AND CONCLUSIONS ... 112
6.5 ELEMENTS IN THE ATMF ... 119
6.6 SUMMARY ... 120

7 RESEARCH STUDY 3: GAME BASED LEARNING ... 121
7.1 INTRODUCTION ... 121
7.2 EUROPEAN CODE WEEK ... 122
7.3 METHODOLOGY ... 123
7.4 RESULTS ... 125
7.5 CONCLUSIONS ... 126
7.6 ELEMENTS IN THE ATMF ... 127
7.7 SUMMARY ... 128

8 RESEARCH STUDY 4: EDUCATIONAL ROBOTICS ... 129

8.1 INTRODUCTION 129
8.2 BACKGROUND 130
8.3 THE RESEARCH 131
8.4 METHODOLOGY 132
8.5 FINDINGS 135
8.6 DISCUSSION 142
8.7 CONCLUSIONS 143
8.8 ELEMENTS IN THE ATMF 144
8.9 SUMMARY 145

9 RESEARCH STUDY 5: PERSONAL LEARNING CHARACTERISTICS 146
9.1 INTRODUCTION 146
9.1.1 Background work and research hypotheses 147
9.1.2 The game activities 151
9.2 METHOD 152
9.2.1 Context 152
9.2.2 Participants 153
9.3 STUDY 1 (H_1): STUDENTS' ATTITUDES TOWARDS SERIOUS GAMES THAT PROMOTE PROGRAMMING (QOLE) 154
9.3.1 Research design 154
9.3.2 Procedure 154
9.3.3 Measures and data analysis 154
9.3.4 Research Findings 155
9.4 STUDY 2 (H_2): CORRELATION OF STUDENTS' QOLE WITH THEIR COGNITIVE STYLE ... 156
9.4.1 Research design 156
9.4.2 Procedure 157
9.4.3 Measures and data analysis 157
9.4.4 Research Findings - Results 157
9.5 STUDY 3 (H_3): CORRELATION OF STUDENTS' PERFORMANCE FROM GAMES WITH THEIR COGNITIVE STYLE 159
9.5.1 Research design 160
9.5.2 Procedure 160
9.5.3 Measures and data analysis 161
9.5.4 Research Findings - Results 162
9.6 ADDITIONAL FINDINGS 162
9.7 DISCUSSION 163
9.8 CONCLUSIONS 164
9.8.1 Guidelines for curricula, game designers and teachers 164
9.8.2 Limitations 166
9.8.3 Future Work 166
9.9 ELEMENTS IN THE ATMF 167
9.10 SUMMARY 169

PART IV - CLOSING 170

10 CONCLUSIONS 171
10.1 SUMMARY OF FINDINGS 171
10.1.1 Characteristics and Strengths of ATM 171
10.1.2 Research studies of this thesis 182
10.1.3 The ATMF 183
10.2 CONTRIBUTIONS AND IMPLICATIONS 184
10.3 FUTURE DIRECTIONS - FINAL REMARKS 185

10.4 THESIS PUBLICATIONS 185
APPENDICES 187
A. EXPERIMENT MATERIALS OF RS 1 187
A.1 Students Test and Questionnaire (in Greek) 187
B. EXPERIMENT MATERIALS OF RS 2 190
B.1 Instructor's Diary of use 190
C. EXPERIMENT MATERIALS OF RS 3 194
C.1 Instructions to parents (In Greek) 194
D. EXPERIMENT MATERIALS OF RS 5 195
E.1 Student Questionnaire (In Greek) 195
E.2 MBTI Questionnaire (In Greek) 199
E.3 Key concepts of the Games used 200
BIBLIOGRAPHY 201

List of Figures

FIGURE 1. A MAP OF THE EMPIRICAL STUDIES IN THIS THESIS AND THEIR CORRESPONDING CHAPTERS. ... 6
FIGURE 2. PCK AS THE INTERPLAY BETWEEN PEDAGOGY AND CONTENT. ... 16
FIGURE 3. THE THREE MAJOR COMPONENTS OF INSTRUCTION. ... 17
FIGURE 4. VYGOTSKY'S ZPD LEARNING THEORY. ... 17
FIGURE 5. BLOOM'S TAXONOMY HIERARCHY (ORIGINAL AND REVISED). ... 19
FIGURE 6. THE FOUR STAGES OF KOLB'S LEARNING CYCLE (ORIGINAL AND UPDATED). ... 20
FIGURE 7. PERRY'S INTELLECTUAL DEVELOPMENT STAGES. ... 20
FIGURE 8. CURRY'S ONION MODEL. ... 21
FIGURE 9. DIAGRAM OF THE LEARNING CYCLE [197]. ... 30
FIGURE 10. SCREEN CAPTURE OF THE CLASSCRAFT GAME. ... 37
FIGURE 11. LEGO MINDSTORMS ROBOTS AND THEIR VPL. ... 39
FIGURE 12. CODING SAMPLE ON A TEXT-BASED LANGUAGE AND ON A VPL. ... 41
FIGURE 13. TOOLS FOR CONTEXTUALIZED PROGRAMMING EDUCATION [366]. ... 44
FIGURE 14. TOOLS FOR MEDIA COMPUTATION [366]. ... 44
FIGURE 15. SORTING NETWORKS - CS UNPLUGGED ACTIVITY. ... 45
FIGURE 16. DEPICTION OF ASCII ENCODER CARD FOR PRACTICE ENCODING SIMPLE DATA, LIKE LETTERS. ... 46
FIGURE 17. EXAMPLE OF LABELING INTO A TEXT BASED PROGRAM. ... 47
FIGURE 18. SUBGOALS IN APP INVENTOR BY VIDEO FORM. ... 47
FIGURE 19. EXAMPLE OF A PAPER-BASED PARSON'S PROGRAMMING PROBLEM. ... 48
FIGURE 20. EXAMPLE OF AN AUTOMATED PARSON'S PROGRAMMING PROBLEM. ... 48
FIGURE 21. EXTREME PROGRAMMING - PRACTICES [232]. ... 49
FIGURE 22. THE FOUR PILLARS OF FLIPPED LEARNING, F-L-I-P TM [424]. ... 54
FIGURE 23. THE FIVE-PHASED METHODOLOGY ADAPTED FOR DEVELOPING THE ATM FRAMEWORK. ... 61
FIGURE 24. THE MODEL OF EDUCATIONAL RECONSTRUCTION AS PROPOSED BY ULRICH KATTMANN AND MICHAEL [488]. ... 63
FIGURE 25. EDUCATIONAL RECONSTRUCTION ACCORDING TO DUIT [490]. ... 63
FIGURE 26. ON THE FUNDAMENTAL INTERPLAY OF INSTRUCTIONAL VARIABLES [491]. ... 64
FIGURE 27. EDUCATIONAL RECONSTRUCTION FOR CSE BY DIETHELM ET AL. [492]. ... 64
FIGURE 28. THE FRAMEWORK FOR TEACHING [30]. ... 65
FIGURE 29. FIRST PRINCIPLES OF INSTRUCTION BY MERRILL [494]. ... 66
FIGURE 30. K-12 CS FRAMEWORK CORE CONCEPTS AND PRACTICES [4]. ... 68
FIGURE 31. BUILDING BLOCKS FOR STANDARDS [4]. ... 69
FIGURE 32. THE CORE PRACTICES OF THE K-12 CS FRAMEWORK AND CT [4]. ... 69
FIGURE 33. METAPHORICAL ILLUSTRATION OF SOLO'S FIVE LEVELS [521] ... 72
FIGURE 34. LEARNING TECHNOLOGY SYSTEMS ARCHITECTURE (LTSA) BY IEEE 1484 [534]. ... 74
FIGURE 35. COMPONENTS END ELEMENTS OF DIMENSION 1. ... 76
FIGURE 36. COMPONENTS END ELEMENTS OF DIMENSION 2. ... 77
FIGURE 37. COMPONENTS END ELEMENTS OF DIMENSION 3. ... 78
FIGURE 38. ALTERNATIVE TEACHING METHODS (SECTIONS OF ELEMENT 2, COMPONENT 3A, DIMENSION 3). ... 78
FIGURE 39. COMPONENTS END ELEMENTS OF DIMENSION 4. ... 79
FIGURE 40. THE ATM FRAMEWORK FOR CSE. ... 80
FIGURE 41. STUDENTS WORKING TOGETHER. ... 88
FIGURE 42. TEACHERS IS JUST WATCHING. ... 88
FIGURE 43. MEANS OF MARKING. ... 91
FIGURE 44. COURSE SATISFACTION. ... 93
FIGURE 45. COURSE DIVERSITY. ... 94
FIGURE 46. COURSE INTEREST. ... 95
FIGURE 47. PERCEIVED KNOWLEDGE. ... 96
FIGURE 48. COURSE ODDNESS. ... 97
FIGURE 49. ATMS USED IN THE RS1. ... 101
FIGURE 50. ATMF ELEMENTS OF RS1. ... 101
FIGURE 51. STUDENTS' RESPONSES ABOUT PRIVACY (I) ON SNS. ... 107
FIGURE 52. STUDENTS' RESPONSES ABOUT PRIVACY (II) ON SNS. ... 108
FIGURE 53. STUDENTS' RESPONSES ABOUT NATURE OF POSTS ON SNS. ... 109

FIGURE 54. GENDER DIFFERENCES IN THE USE OF SNS. 109
FIGURE 55. ACADEMIC AND REAL LIFE GAP (I) WITH FACEBOOK. 110
FIGURE 56. ACADEMIC AND REAL LIFE GAP (II) WITH FACEBOOK. 110
FIGURE 57. COURSE EVALUATION WITH FACEBOOK. 111
FIGURE 58. PRIVACY EXAMPLES ON SNS. 112
FIGURE 59. NEW FRIENDSHIPS ON FACEBOOK. 112
FIGURE 60. REACTION TO ART VIDEO ON COURSE'S FACEBOOK PAGE. 113
FIGURE 61. REACTION TO BIRTHDAY WISHES ON COURSE'S FACEBOOK PAGE. 113
FIGURE 62. REACTION TO HUMORISTIC PHOTOS ON COURSE'S FACEBOOK PAGE. 113
FIGURE 63. REACTION TO HUMORISTIC VIDEO ON COURSE'S FACEBOOK PAGE. 114
FIGURE 64. COURSE WORK EVALUATION IN THE FORM OF OSCARS. 116
FIGURE 65. EVALUATION ON WORK IN PROGRESS. 116
FIGURE 66. ANNOUNCING EVENTS ON COURSE'S FACEBOOK PAGE. 117
FIGURE 67. ATMS USED IN THE RS2. 119
FIGURE 68. ATMF ELEMENTS OF RS2. 119
FIGURE 69. THE GREEK POSTER OF CS TEACHERS' COUNSELORS. 122
FIGURE 70. THE UNIVERSITY'S POSTER ABOUT THE CODING EVENT. 123
FIGURE 71. PUPILS WORKING TOGETHER. 123
FIGURE 72. UNDERGRADUATE STUDENTS HELP THE LITTLE PUPILS. 124
FIGURE 73. PUPILS SHOWING THEIR CERTIFICATES. 124
FIGURE 74. TEACHER'S DASHBOARD SHOWING PUPILS' PROCESS. 125
FIGURE 75. ATMS USED IN THE RS3. 127
FIGURE 76. ATMF ELEMENTS OF RS3. 128
FIGURE 77. ATM USED IN THE RS4. 144
FIGURE 78. ATMF ELEMENTS OF RS4. 145
FIGURE 79. CONCEPTUAL MODEL OF THE RS5. 147
FIGURE 80. SCREENSHOTS FROM CODE.ORG® GAMES. 151
FIGURE 81. GRAPHICAL REPRESENTATION OF THE RESEARCH DESIGN OF THE 1ST STUDY. 154
FIGURE 82. PARTICIPANTS' SATISFACTION AND EASE OF USE. 155
FIGURE 83. PARTICIPANTS' PERCEIVED LEARNING AND INTENTION TO PARTICIPATE IN SIMILAR ACTIVITIES. 156
FIGURE 84. PARTICIPANTS' MOTIVATION. 156
FIGURE 85. GRAPHICAL REPRESENTATION OF THE RESEARCH DESIGN OF THE 2ND STUDY. 157
FIGURE 86. CROSSTAB - S-N COGNITIVE DIMENSION WITH EOU AND LEA. 158
FIGURE 87. CROSSTAB - S-N COGNITIVE DIMENSION WITH GBL MOTIVATION. 159
FIGURE 88. CROSSTAB - J-P COGNITIVE DIMENSION WITH LEA AND MOT. 159
FIGURE 89. GRAPHICAL REPRESENTATION OF THE RESEARCH DESIGN OF THE 3RD STUDY. 160
FIGURE 90. PARTIAL VIEW OF TEACHER'S DASHBOARD (HOUR OF CODE 2013). 160
FIGURE 91. CROSSTAB – COGNITIVE STYLE WITH GAMES ACHIEVEMENT. 162
FIGURE 92. ATMS USED IN THE RS5. 168
FIGURE 93. ATMF ELEMENTS OF RS5. 168
FIGURE 94. THE FRAMEWORK OF THIS THESIS (ATMF). 183

List of Tables

TABLE 1. COMPUTING EDUCATION TAXONOMY USED IN THIS THESIS. 13
TABLE 2. CS KNOWLEDGE AREAS IN HIGHER EDUCATION ACCORDING TO ACM. 14
TABLE 3. STEPS INVOLVED IN PBL. 26
TABLE 4. DIFFERENCES BETWEEN COOPERATIVE AND COLLABORATIVE LEARNING [222, 223]. 33
TABLE 5. COMPARISON OF GAMIFICATION AND GBL. 37
TABLE 6. EDUCATIONAL ROBOTIC PLATFORMS. 39
TABLE 7. VISUAL PROGRAMMING ENVIRONMENTS FOR EDUCATION. 41
TABLE 8. DIFFERENCES BETWEEN TRADITIONAL AND BLENDED LEARNING [417]. 53
TABLE 9. STUDENTS BY GENDER AND LEARNING GROUP. 85
TABLE 10. TEAMS' OVERALL TEACHING SCORES. 89
TABLE 11. TEST SCORE - DESCRIPTIVE STATISTICS. 90
TABLE 12. TEST OF HOMOGENEITY OF VARIANCES. 91
TABLE 13. TEST SCORE - ANOVA. 91
TABLE 14. ROBUST TESTS OF EQUALITY OF MEANS. 92
TABLE 15. TEST STATISTICS[B] FOR SATISFACTION. 92
TABLE 16. DESCRIPTIVE STATISTICS FOR SATISFACTION. 92
TABLE 17. MEAN RANKS FOR SATISFACTION. 92
TABLE 18. TEST STATISTICS[B] FOR DIFFERENCE. 93
TABLE 19. DESCRIPTIVE STATISTICS FOR DIFFERENCE. 93
TABLE 20. MEAN RANKS FOR DIFFERENCE. 94
TABLE 21. TEST STATISTICS[B] FOR INTEREST. 94
TABLE 22. DESCRIPTIVE STATISTICS FOR INTEREST. 94
TABLE 23. MEAN RANKS FOR INTEREST. 95
TABLE 24. TEST STATISTICS[B] FOR LEARNING. 95
TABLE 25. DESCRIPTIVE STATISTICS FOR LEARNING. 96
TABLE 26. MEAN RANKS FOR LEARNING. 96
TABLE 27. TEST STATISTICS[B] FOR ODDNESS. 96
TABLE 28. DESCRIPTIVE STATISTICS FOR ODDNESS. 97
TABLE 29. MEAN RANKS FOR ODDNESS. 97
TABLE 30. LEVELS OF BLOOM'S TAXONOMY USED IN THIS STUDY. 99
TABLE 31. PROFILE OF THE 18 PARTICIPANT TEACHERS IN THE ER COMPETITION. 133
TABLE 32. MOTIVATION: MOST COMMONLY OCCURRING THEMES. 136
TABLE 33. SKILLS: MOST COMMONLY OCCURRING THEMES. 137
TABLE 34. PROGRAMMING: MOST COMMONLY OCCURRING THEMES. 138
TABLE 35. ENGINEERING: MOST COMMONLY OCCURRING THEMES. 139
TABLE 36. CURRICULUM: MOST COMMONLY OCCURRING THEMES. 140
TABLE 37. OTHER ROBOTS: MOST COMMONLY OCCURRING THEMES. 141
TABLE 38. FUNDING RESOURCES: MOST COMMONLY OCCURRING THEMES. 141
TABLE 39. DIFFICULTIES: MOST COMMONLY OCCURRING THEMES. 142
TABLE 40. QUALITY OF LEARNING EXPERIENCE FOR THIS STUDY AND EXPLANATION 149
TABLE 41. STATISTICS ON STUDENTS' QUALITY OF LEARNING EXPERIENCE. 155
TABLE 42. MBTI DIMENSIONS. 157
TABLE 43. PEARSON'S CORRELATION COEFFICIENT BETWEEN MBTI DIMENSIONS. 158
TABLE 44. CALCULATION FOR STUDENTS' PERFORMANCE FROM GAMES. 160
TABLE 45. GBL ASSESSMENT THROUGH THE PUZZLE GAMING ACTIVITIES. 161
TABLE 46. SIMPLIFIED STUDENTS' OVERALL PROGRESS FROM GAMING ACTIVITIES. 161
TABLE 47. CHI SQUARE ANALYSES WITH FURTHER FINDINGS REGARDING COGNITIVE STYLES. 163
TABLE 48. ALTERNATIVE TEACHING METHODS THAT HAVE BEEN USED IN CSE. 172

Part I - Introduction

Chapter 1

Teaching is the highest form of understanding

Aristotle

Introduction

Informatics or else named Computer Science (CS) is on the brink of an enormous possible growth and therefore it draws great interest from governments, businesses and other organizations as a top educational priority. In September 2013, the University of Chicago's Center for Elementary Mathematics and Science Education (CEMSE) and Urban Education Institute (UEI) worked with a partnership established by the Association for Computing Machinery (ACM)[1] and completed a collaborative research to establish a more comprehensive understanding of Computer Science Education (CSE), entitled "Building an Operating System for Computer Science Education" [1]. In January 2016, president Obama of the United States of America announced a new initiative called "Computer Science for all" [2], to empower all students from kindergarten through high school to learn CS and to be equipped with computational thinking (CT) skills, in order to be creators in the digital economy, not just consumers, and to be active citizens in our technology-driven world [3]. Following, in October 2016, ACM together with Computer Science Teachers Association[2] (CSTA) and Code.org[3] announced the development of a framework for CSE [4], in order to make clear what should all students know and what they should be able to do within CS.

What do these movements have in common? It seems that there is a need for greater access to CSE and in response, computer scientists are trying to fill the gap, in an area with far too little research [5], in comparison to well established fields such as Mathematics [6].

1.1 Research Motivation

The dynamic development of the CS field introduces new educational and pedagogical challenges, including the instructional design of teaching and learning [7]. How can we teach our students better in such a growing and demanding field? Moreover, how can we motivate them and together have better learning results? The answers to those questions are not simple, but it is clear that new methods of teaching and learning are needed [8].

The issue of finding better ways to teach Informatics came at the center of attention when a stark report entitled "Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age" [9], revealed several issues in CSE, like the lack of CS standards and more effective teaching ways. In this perspective, the "ACM K-12 Education Task Force Report" draws attention to the need for appropriate CS teacher training programs [10] and reports that "teachers must acquire both a mastery of the subject matter and the pedagogical skills that will allow them to present the material to students at appropriate levels" [11]. Yet, Lapidot and Hazzan, refer to different topics that should be researched further in CSE, like pedagogical approaches for teaching different subjects and tools for assessing students' performance [12, 13]. They also emphasize the need to use new teaching

[1] Association for Computing Machinery (ACM): https://www.acm.org/, last accessed December 2016
[2] Computer Science Teachers Association (CSTA): http://www.csta.acm.org/, last accessed December 2016
[3] Code.org: https://code.org/, last accessed December 2016

approaches with active learning and find effective ways to integrate them into classroom settings [13, 14].

As recent efforts in several countries show, the interest in CSE is growing with no boundaries. In the USA for example, the number of undergraduates declaring a computing major at Ph.D. has increased 60% from 2011-2014 [15]. This growth is not limited to university education. Many big cities, like New York, San Francisco and Oakland will soon be offering CS to all students at all public schools from preschool to high school students [8]. In Europe similarly, the subject of CS is also gaining high importance. The European Commission has classified coding as the "skill of the 21st century"[4]. In UK, the Royal Society has launched the initiative entitled "Shutdown or Restart" [16] to improve CSE at schools, while the Computing at School (CAS) working group, promotes the teaching of foundational concepts at every school. Denmark has begun national computing curricula [17]. In Spain many regions recently introduced CS in the curriculum while at the other end of the world the New Zealand's ministry of education recently introduced a new CS subject in every high school [18, 19]. Moreover, hundreds of thousands of learners are enrolling in MOOCs (Massive Open Online Courses) on computing and millions more are learning through online tutorials, such as those provided by Codecademy.org[5] and Code.org[6]. This unparalleled demand means that schools and universities are going to teach not only more students in the coming decades, but also more diverse students, with more varied backgrounds, motivations, preparations, and abilities. This a challenging aim since we know very little about how best to teach our existing students, let alone the upcoming growth of them [8]. However, this challenge also presents a great opportunity to expand the influence of CSE to more students.

Meeting this goal requires training thousands of new CS teachers [20-22]. One possible way is to develop and deploy strategies for existing teachers with more student-centered orientation reclaiming practices from other content areas to become effective CS teachers through professional development [1]. However, most instructors have little practice or experience with teaching methods other than traditional lecturing [23]. Some have a particular teaching style that they use in every teaching context, regardless of the type or level of student learning they expect. Others, because of the CS discipline, may already use various technology applications, but their decisions may not be based on student-centered pedagogy [24, 25]. Finally, a number of instructors are concerned that students will respond negatively to teaching and learning activities that are new to them [26]. Thus, any of these situations can make it hard to reframe teaching to a student-centered perspective. It is worth mentioning that most lecturers in higher education employ traditional, lecture-based instruction, essentially following the teaching that was modeled for them in their own university studies [27-29]. It becomes apparent that there is a need for a clear, student-centered framework encompassing alternative teaching concepts.

One may argue that several models exist, standards, curriculum guidelines or frameworks that can be used for teaching and learning purposes. However, most of them refer to teaching in a general scope [30] or teaching in a specific discipline like frameworks from math, science, and technology education [31, 32]. Therefore, the existing frameworks do not give emphasis to the identity of CSE [33] and moreover to alternative teaching strategies. For example, within the CS field, the "K–12 Computer Science Framework", describes a foundational literacy in CS, aiming to show that CS is

[4] Coding is the 21st century skill: https://ec.europa.eu/digital-single-market/en/coding-21st-century-skill, last accessed December 2016
[5] Codeacademy: https://www.codecademy.com/, last accessed December 2016
[6] Code.org®: https://code.org , last accessed December 2016

essential for all students [4]. A framework that includes standards, curriculum, course pathways and even professional development suggestions for all K–12 grade levels, but does not include specific teaching guidelines. Yet, other national frameworks from United Kingdom [34], Germany [35], Poland [36] and New Zealand [19], describe CS concepts and practices that students should know, and do not focus on alternative teaching methods. Finally, other well established frameworks in higher education, like the Advanced Placement CS Principles curriculum framework [37] and the ACM's curriculum guidelines for undergraduate CS programs [38], provide standards for students and not for teachers who want to teach students.

Additionally, although a teaching framework could be based on existing research and well established practices, it should also be evolving, taking into account new empirical studies. Hubwieser [39] emphasizes that all choices of any country that aim to improve CSE should be made on the basis of empirically proven facts instead of personal beliefs or suggestions. The lack of experiential research on best practices in CSE has led researchers to repetitively ask similar basic questions without clear progress toward resolving them, although current practical researches show promise [40, 41]. Moreover, in Greece research activities and empirical evidences about teaching and learning with alternative methods in the CSE field are limited [42, 43].

The challenges that I consider in this thesis are:

- to research and describe every known teaching method that is used in CSE;
- to provide qualitative and quantitative means with use of some alternative teaching methods in Greek educational settings;
- to develop an educational framework for the use of alternative teaching methods and strategies in CSE.

1.2 Research Goal and Questions

Given the research motivations discussed in the previous section, it is clear that there is potential for learning achievement to be improved in CSE through the effective design and use of new methods, strategies and approaches. However, it is argued that not many active learning approaches have been evaluated in Greek educational settings [42].

Hence, the main goal is to design an educational framework based on alternative teaching methods and test it in the Greek education. A student-centered framework for teaching can provide shared understandings, which can help improve the quality of instructional design, course and lesson planning, learning and assessment.

The research questions that this thesis aims to answer are as follows:

Research Question 1: What are the alternative teaching methods that can be used in CSE, to complement and enhance the existing (traditional) teaching and learning?

Research Question 2: What are the key benefits of implementing alternative teaching methods in Greek CSE?

Research Question 3: How can alternative teaching methods be used effectively in CSE of any context, level and subject?

To my knowledge, a review that directly addresses these questions does not previously exist. There are broader classifications of methods in CSE, such as tools for introductory programming education [44], but this thesis seeks a slight difference view of CSE. This thesis produces classifications that can serve as a map of the terrain and looks deeper into alternative teaching within CSE. The above research questions are established through the literature review in Chapters 2, 3 and 4. In particular, the first research question is investigated in Chapter 3. The second research question is investigated in Chapters 5, 6, 7, 8 and 9. Finally, the third research question is investigated in Chapters 4, 5, 6, 7, 8 and 9.

1.3 Definitions

This thesis uses several terms that will be defined here for clarification. First, both Computer Science and Informatics will be referred to as CS throughout the paper for brevity and convenience - with the context clearly described in chapter 2. Moreover, Computer Science Education will be referred to as CSE and Alternative Teaching Methods as ATM.

Additionally, the participants of the studies described in this thesis are addressed by using the following terminology:

- Learners: CS learners in any context, either at university or primary, middle and high school level.
- Students: High school learners. They come from two kinds of schools in Greek education; junior high school and senior high school (called Gymnasium and Lyceum respectively).
- Pupils: Elementary education learners.
- Instructors: Instructors either in the university or in the primary, middle and high school e.g. teachers, professors.

Finally, throughout the thesis the term “school” is used in its Greek sense, to mean primary and secondary school, ages 6-18, corresponding to K-12 at USA.

1.4 Thesis Outline

Figure 1 provides a graphical overview of this thesis, depicting the empirical studies conducted in Greek educational settings to answer the research questions above. The studies described in Chapters 5 through 9, were all previously reported elsewhere in peer-reviewed publications as described in 10.4. The model in Figure 1 consists of three interrelated levels: Initially are presented the chapters that each empirical study is reported aside with the main characteristics that it researches, like learning and motivation. Next, is shown the alternative teaching approach that was implemented for each study. Finally, at the bottom of this visualization, are presented the proposed framework’s stages corresponded with the previous approaches and studies.

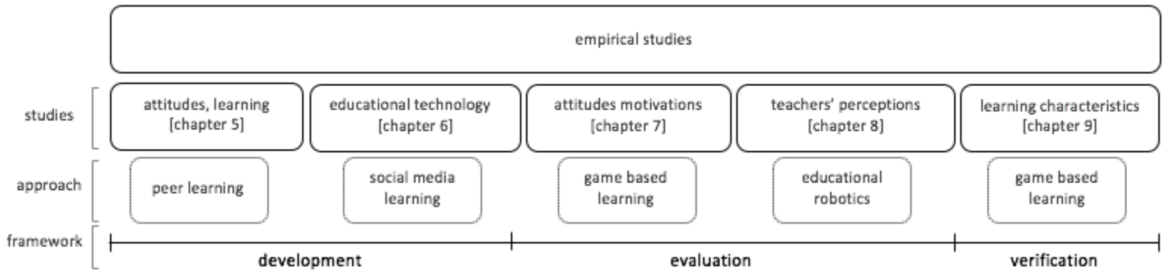


Figure 1. A map of the empirical studies in this thesis and their corresponding chapters.

Moreover, this thesis is organized in four parts consisted of ten chapters as follows:

- **Chapter 2** - Background: the discipline
  This chapter summarizes topics encountered in CSE. It highlights the characteristics that determine CS and what researchers have found interesting enough to pursue to create the field.
- **Chapter 3** - Background: Teaching and Learning
  This chapter provides an outline of learning theories and teaching methods used in CSE. Extensive literature review about alternative teaching methods and their benefits is presented.
- **Chapter 4** - The Framework
  This chapter presents the Framework developed through this thesis. It describes other teaching and learning frameworks and illustrates the alternative teaching practices in CSE.
- **Chapter 5** - Research Study 1: Peer Learning
  This chapter investigates alternative teaching methods in Greek Secondary education. It presents results from Peer Learning and Collaboration techniques and contributes to the proposed framework.
- **Chapter 6** - Research Study 2: Social Networks in Education
  This chapter investigates Social Network Learning as an assisted tool. It presents findings from a case study about the use of Facebook in higher education and adds characteristics to the proposed framework.
- **Chapter 7** - Research Study 3: Game Based Learning
  This chapter investigates the method of Game Based Learning in primary education. Serious games that promote algorithmically thinking were used in this observational study and pupils' attitudes were recorded. The study was used to refine the proposed framework.
- **Chapter 8** - Research Study 4: Educational Robotics
  This chapter investigates the use of Educational Robotics as an alternative teaching method. It presents qualitative data from teachers that used the method during a competition. The study was used to refine the proposed framework.
- **Chapter 9** - Research Study 5: Personal Learning Characteristics
  This chapter investigates the Game Based Learning in secondary education students. This time serious games were used to find potential differences between learning, motivation and different personalities. The study was used to finilize the proposed framework.
- **Chapter 10** - Closing: Conclusions

Summarizes the thesis' main achievements and contributions and discusses current limitations. Moreover, future directions and potential solutions that might embrace these drawbacks are presented.

1.5 Summary

Mainly motivated by the enormous growth of CSE, the current lack of effective teaching and learning methods, this dissertation covers the following issues: It presents alternative teaching methods approaches and strategies that are used in CSE; test some of them in the Greek education primarily in terms of motivation and learning; develops a framework to teach CS concepts through alternative teaching methods in a way that appeals to a broad audience, engages its users, and shows measurable learning outcomes. This chapter presented the motivation for writing this thesis, the research problem and the research goals and questions that is trying to answer. Finally, the structure of the thesis is presented.

Part II - Mapping the territory

Chapter 2

I hear and I forget,
I see and I remember,
I do and I understand.
Confucius

The Discipline

This chapter is mainly a work of cartography. Its aim is to map a territory of topics encountered in CSE. Therefore, this map presents what CS stands for and distinguishes it from close related fields. It also reports some historical aspects, in that it represents what researchers have found interesting enough to pursue to create the field. Then, it highlights the characteristics that determine the discipline and the special skills that learners can develop. Finally, it summarizes all the above into a taxonomy with the core elements that should taught within all levels of CSE.

2.1 Defining the Terminology

CSE is learning about CS and focuses on teaching the fundamental concepts of the discipline, just as core Mathematics and Physics courses do. CS has its roots mainly in mathematics but also in technical and engineering sciences [45-47]. Yet, the are several answers to the question 'what is computer science?'. Different universities, researchers, practitioners and countries emphasize many different aspects of the field. For the purposes of this thesis, we rely on the definition provided in the ACM/CSTA Model Curriculum for K–12 Computer Science: "*CS is the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society.*" [48]. This definition presents algorithms as the main object investigated within CS. However, it neglects to properly reveal the nature and methodology of CS.

In a broader view, CS is part of the Computing field, which also includes computer engineering, software engineering, information systems, and information technology. The following definitions help distinguish the scientific discipline of CS from other terms commonly used to describe the use of computers in education:

Information and Communications Technology (ICT)

ICT is an umbrella term that includes any communication device or application. It is the study of the technology used to handle information and aid communication. This extended term stresses the role of unified communications and the integration of telecommunications and computers. ICT can enhance teaching and learning through its dynamic, interactive, and engaging content [49].

Technology literacy and fluency

A range of curricula going understanding how to use technology to the ability to express ideas creatively, reformulate knowledge, and synthesize new information and technology.

Educational technology (ET)

ET has to do with the integration of technology into teaching in order to advance student learning. It focuses on the use of computing as a tool to solve problems in other fields than CS, specifically the use of computing applications in pursuit of that goal.

Information technology (IT)

IT is a broad and diverse set of topics, but typically focused on applying the components of computing to the acquisition and/or analysis of information, in order to solve a business information problem, such as network or database administration.

Computing education:

It is a broadly used term that can encompass some or all of the terms noted above. The ACM Computing curricula defines it as: "*any goal-oriented activity requiring, benefiting from, or creating a mathematical sequence of steps known as an algorithm; e.g. through computers. Computing includes designing, developing and building hardware and software systems; processing, structuring, and managing various kinds of information; doing scientific research on and with computers; making computer systems behave intelligently; and creating and using communications and entertainment media.*" [50].

Computing curricula used in education and trends that have occurred have changed over the last decades in CSE. In the beginning of the 1980s, the CSE was established in the majority of schools of many countries, e.g. Austria, Germany, Lithuania, and Russia. Furthermore, several major organizations have developed curriculum guidelines from kindergarten to university education. ACM and IEEE-CS joined their forces in the late 1980s to create a curriculum report for computing. Published in 1991 and known as Computing Curricula 1991 [51], it provided guidelines for undergraduate programs in CS and computer engineering. Later, in 1993, ACM produced curriculum recommendations for a high school curriculum [52]. By the end of the 1990s, it was becoming clear that the computing field had grown rapidly and in many dimensions. Therefore, ACM and IEEE-CS created again a joint task force and to produce Computing Curricula 2001 (CC2001) [53], aiming to the various computing disciplines. That report included CS curriculum guidelines, but it mainly presented the richness and the range provided by the various computing disciplines. In response to the CC2001 standard, other discipline-specific volumes like software engineering [54] and computer engineering [55] developed curriculum reports.

Nevertheless, a confusion arises when trying to distinguish between the most common areas of computing education offered in schools like CS, IT and ET [56]. The need for a basic understanding of the discipline and what should CSE include, seems to be common. The CSTA in their "K-12 CS standards report" [56] argues that every CS course should have the following characteristics:

- CS courses must begin at the elementary school level by introducing the fundamental concepts to all students.
- They should present CS at the secondary school level, in a way that seems to be both accessible and worthy of an academic curriculum credit (e.g., math or science).
- Should offer additional secondary level CS courses, in order to allow interested students to study it in depth and prepare them for entry into the work force or higher education, and
- Should increase the knowledge of CS for all students.

However, just as important as recognizing what CS is, it is also recognizing what it is not. There is a growing body of work being done with/on computers in education, both from computer scientific as well as from pedagogical and sociological points of view [57, 58]. This kind of work is different from CSE in that this it does not usually focus on the subject matter of the discipline, but on the implementation of technology as a pedagogical means in the teaching and learning of other subjects.

Yet, a separation should also be made between the teaching and learning of CS as vocational training on the one hand and as a theoretical academic study on the other. The focus of this thesis is on the teaching and learning of the discipline of CS, purely for theoretical purposes.

Moreover, CSE has merged with the science policy attention to Science, Technology, Engineering, and Mathematics learning (STEM), since the turn of the 21st century. In a recently report [59] about STEM education in the USA, authors point out that the most important STEM field is CS and that for a "modern economy", it lacks its own initial in STEM and also has the fewest number of high school students taking its classes. According to the report, skills in CS don't exist just in IT professions, but in all segments. Furthermore, CT is viewed as at the core of all STEM characteristics [60]. Concerns about CS within STEM education deepen also from predictions given from the Bureau of Labor Statistics in the USA[7], that computing is one of the fastest-growing job markets through 2020. Many also assume that CS is the "T" in STEM, but this is not the case. The "T" in STEM is a diverse space that includes many aspects, only some of which may relate to computing and CS. Yet, Wilson and Guzdial [61] report that although there have been funded huge amounts of money for strengthening STEM in K–12 schools of USA, research explicitly in computing remains underfunded. However, CS with or without STEM concepts is increasingly important to students since it provides numerous essential skills for the modern world, as described in the following section.

2.2 Skills and Characteristics

As reported in the previous section, there are many different aspects of CSE. To the same reason, there is an ignorance about the core concepts and characteristics that students are taught through CSE. This reflects to the wider ignorance by the general public about the nature, methodologies, and contributions of the field in the modern world. Moreover, the computing sub-disciplines have much in common as well as distinguishing characteristics [62]. For this reason, in 2007, ACM launched the "Computing Ontology Project" [63] in order to organize and identify commonalities and differences that exist within computing disciplines in a formal manner. The project describes and analyses various characteristics of computing education. However, there are specific fundamental ideas that learners can acquire throughout their studies in the CS field [64]. This section presents the main concepts and characteristics that CS promotes in education.

The core characteristic in CSE is programming (or sometimes coding). Characteristically, Ershov, one of the early Russian pioneers in the theoretical CS field, stated that: "*Programming is the second literacy*" [65]. Teaching CS started with programming and that slogan has become a popular metaphor, which is used widely around the world. Programming is about the process of writing programs. Programs may have various looks like text or shapes and in they can have a variety of instructions that command a computer to behave in a certain way. That behaviour is the process and a program specifies a computational process which can be executed at some time [66].

Computer programming skills enhance computational thinking (CT). Jeannette Wing shaped the term in 2006 [67], and she argued that CT involves concepts like "*problem solving, designing systems, and understanding human behavior*". In addition, it is correlated with analytical thinking sills, since it shares with mathematical thinking in the general way in which someone approaches solving a problem. It

[7] Bureau of Labor Statistics in the United States of America http://www.bls.gov/ooh/, last accessed December 2016

also shares with engineering thinking in the way in which someone approaches designing and evaluating a large, complex system that operates within the constraints of the real world. It finally shares with scientific thinking in the way in which someone approaches understanding computability, intelligence, and therefore the human mind behavior. Therefore, CT is considered a fundamental ability for the 21st century, for everyone and not just the computer scientists [68].

Programming is not only a fundamental skill of CS and a key tool for supporting the cognitive tasks involved in CT, but is also a demonstration of computational competencies as well. Significant efforts like CS Unplugged[8] that introduce computing concepts without the use of computers, while providing valuable introductory activities for exposing children to the nature of CS, may be keeping learners from the crucial computational experiences involved in CT's common practice. However, as mentioned the goal of teaching programming is problem-solving transfer, i.e., users are expected to be able to apply what they have learned to solve problems that they have not been taught. The first step in solving a problem is to state it clearly and unambiguously and CS teaches students to think about the problem-solving process itself. Students learn how to think differently about problems and try to solve them in any context.

Furthermore, learning programming means learning a language of communication with machines [69]. According to Hromkovic [69] in order to tell a device what activity we want to have from it we have to describe all possibly complex behaviors by a sequence of clear, simple instructions, therefore to teach programming as a skill. Furthermore, the ability to use computers to express ideas and to consume others' ideas is known as Computational Literacy [66]. In order someone to be able to use a computer expressively she must know it's capabilities and limitations. And achieving computational literacy means that someone can read and write with computation, which includes the ability to read and write computer programs.

In addition, CS teaches the way to create digital artifacts and how those artifacts can impact the world around them by looking at issues such as privacy and security. A simple way to think about this is that it teaches kids how to create new technologies instead of just being consumers of technology [3].

It is obvious that through CS learners can acquire many valuable skills and characteristics. Nevertheless, almost since its beginning, it has been followed by the perception that it focuses exclusively on programming skills. Coding is an important tool for CS but it is a bit like arithmetic is a tool for doing mathematics, and words are tools for Greek. Coding can create software, but CS is a broader field covering many different concepts that go well beyond coding. Through CS, one can develop logical reasoning and can gain awareness of the resources required to implement, test, and deploy a solution and how to deal with real-world constraints. All these skills are applicable in many contexts, from science and engineering to the humanities and business, and they have enabled deeper understanding in these and other areas.

2.3 Taxonomy for CSE

As, shown at the previous sections of this chapter, there is some confusion around the explosion of terms used to describe the various kinds of computing education. From a pedagogical perspective, according to Ragonis [70], there is a lack of standardization in the interpretation that various groups

[8] CS Unplugged web page: http://csunplugged.org/, last accessed December 2016

and countries give to curricula in CS. Different approaches, appear with different names in the curricula, such as IT, CS, informatics, computer engineering, software engineering and even ET. There is also a need to improve the level of CS understanding as an academic and professional field, including its distinctions from other sciences like mathematics and also from close related fields like IS and IT. This section presents a taxonomy (Table 1) based on standards from ACM, IEEE and CSTA and highlights the principles that are more suitable to be taught at each educational level. This classification system is ordered in different computing fields and levels and it describes and categorizes different characteristics of CSE.

Table 1. Computing Education Taxonomy used in this thesis.

<table>
<tr><td colspan="10">1. ICT & Education
(the fields)</td></tr>
<tr><td>1.1 Computer Science (Informatics)</td><td colspan="2">1.2 Technology literacy and fluency</td><td colspan="2">1.3 Educational Technology or Computing across the curriculum</td><td colspan="2">1.4 Information Technology</td><td colspan="3">1.5 Computing Education</td></tr>
<tr><td colspan="10">2. CSE learning
(levels & concepts)</td></tr>
<tr><td colspan="3">2.1 Elementary/Middle School</td><td colspan="2">2.2 High School</td><td colspan="5">2.3 College/University</td></tr>
<tr><td>2.1.1 Problem Solving</td><td colspan="2">2.1.2 Computational Thinking</td><td>2.2.1 Computing</td><td>2.2.2 Computer Science</td><td colspan="2">2.3.1 Enrollment</td><td colspan="3">2.3.2 Transition</td></tr>
<tr><td colspan="10">3. CSE teaching
(education level & characteristics & concepts)</td></tr>
<tr><td colspan="5">3.1 Complementary and Essential Elements in Primary & Secondary Education</td><td colspan="5">3.2 Computing Education Curricula in Higher Education</td></tr>
<tr><td>3.1.1 Computational Thinking</td><td>3.1.2 Collaboration</td><td>3.1.3 Computing practice & Programming</td><td>3.1.4 Computers & Communication devices</td><td>3.1.5 Community, Global & Ethical Impacts</td><td>3.2.1 Computer Science *</td><td>3.2.2 Computer Engineering</td><td>3.2.3 Information Systems</td><td>3.2.4 Information Technology</td><td>3.2.5 Software Engineering</td></tr>
</table>

The CS2013 Body of Knowledge [71] is organized into a set of 18 Knowledge Areas (KAs), corresponding to topical areas of study in computing (Table 2).

Table 2. CS Knowledge Areas in higher education according to ACM.

* 3.2.1 Computer Science KAs																	
AL - Algorithms and Complexity	AR - Architecture and Organization	CN- Computational Science	DS - Discrete Structures	GV - Graphics and Visualization	HCI - Human-Computer Interaction	IAS - Information Assurance & Security	IM - Information Management	IS - Intelligent Systems	NC - Networking and Communications	OS - Operating Systems	PDB - Platform-based Development	PD - Parallel and Distributed Computing	PL - Programming Languages	SDF - Software Development Fundamentals	SE - Software Engineering	SF - Systems Fundamentals	SP - Social Issues and Professional Practices

2.4 Summary

This chapter maps a territory of topics encountered in CSE. Firstly, it reports historical aspects of the CS field and presents what researchers have found interesting enough to pursue to create it. Then, it highlights the characteristics that determine the discipline and the special skills that learners can develop through it. Finally, it summarizes all the above into a taxonomy (based on standards from CSTA and ACM) with the core elements that should taught within all levels of CSE. The following chapter illustrates comprehensively the learning methodologies used in CSE and introduces the first step towards applying the proposed techniques for creating an alternative teaching framework.

Chapter 3

> *Nothing has brought pedagogical theory into greater disrepute than the belief that it is identified with handing out to teachers' recipes and models to be followed in teaching.*
> *John Dewey*
> *Democracy and Education*

Teaching and Learning

In a very general level, teaching is about transmitting knowledge while learning involves acquiring and modifying knowledge. This chapter highlights certain teaching and learning issues, like theories, models and taxonomies that are well known in the field of CSE. The first section presents characteristics regarding the teaching in CSE. Next, learning theories are presented. Then all the known teaching strategies, approaches and methods that are used within CSE, traditional and alternative are reported.

3.1 The Teaching

Teaching is a complex, multi-layered activity, which requires instructors to manipulate multiple tasks and goals simultaneously. CS teaching became an autonomous scientific field, known as instructional (or didactic) research, the last three decades [72]. Its purpose is the study of the construction of knowledge, the development of intellectual and technical skills, attitudes and values of all education partners engaged in CS [33].

But what knowledge do teachers need to have for teaching? Responses to this question have evolved over the past centuries, with significant changes at the beginning of the 20th century [73]. Up through the 19th century, the prevailing notion was that teachers needed to know the content they were to teach. This view shifted to the importance of knowing how to teach. Teachers should implement new teaching and learning (or pedagogical) practices along with an even more in depth understanding of the content they were planning on teaching [73, 74]. The late 80s Shulman [75] challenged educators and researchers to reconsider the knowledge that they need to know so as to be the best possible teachers. He supposed that, a teacher along with mastering the subject matter of her discipline well, she must also include the following:

- Content knowledge.
- General pedagogical knowledge.
- Curriculum knowledge.
- Pedagogical content knowledge.
- Knowledge of learners.
- Knowledge of educational contexts.
- Knowledge of educational ends, purposes, and values.

The Pedagogical Content Knowledge (PCK), was the category most likely to distinguish the understanding of the content specialist from that of the pedagogue [76]. PCK refers to the overlap of information about subject knowledge, that is knowledge of the subject being taught, and pedagogic knowledge, that is knowledge of how to teach (Figure 2). Shulman [75] conceptualized it by including conceptual and procedural knowledge, a range of varied techniques or activities, knowledge of methods for evaluation and knowledge of a variety of resources which can be easily accessed. He claimed that teachers have a unique way of looking at practice and encouraged the examination of their pedagogical thinking in ways that would reveal what they must know to best teach the content to their students. Therefore, PCK is an approach to thinking about instructor's knowledge and is considered as an essential part of the educational research [77].

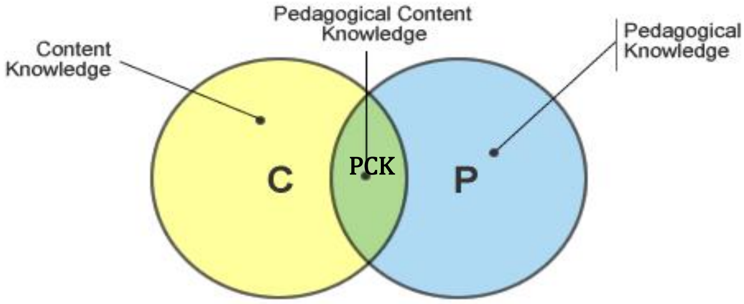


Figure 2. PCK as the interplay between pedagogy and content.

An effective way for increasing CS teachers' PCK is to provide them with opportunities tied to their curricular needs, develop their content knowledge, develop their understanding about students' needs and furthermore engage them in continuous reflection and dialogue about teaching [78, 79]. However, we know very little about how CS teachers come to develop these knowledge methods and what challenges they face to support their identity development [20, 22].

Saeli et al. [80] aimed to reveal PCK within the context of learning programming. They asked four core questions: what are the reasons to teach programming; what are the concepts we need to teach programming; what are the most common difficulties/misconceptions students encounter while learning to program; and how to teach this topic. Authors found that responses to those questions were not connected to each other and recommended additional work in order to represent programming PCK both from a general perspective as well as most frequently taught topics. In another study, Liberman et al. [81] explored the relationship between CS teachers' content knowledge and PCK as they shifted from procedural programming to object- oriented programming. The authors found that there is a state when teaching (called regressed state), where even more experienced CS teachers incorporated elements that are more typical to novice teachers.

Moreover, research in other disciplines offers some insight into how CS teachers without sufficient background come to face teaching challenges inside classroom. Previous work has proposed that beginning teachers face a number of challenges, including: teaching multiple subject areas, classroom management, and insufficient planning time [82-84]. Veenman [82] conducted a meta-analysis of 83 studies that focused on problems that teachers encounter within classroom settings. By reviewing the studies, the author identified the most perceived problems including lesson planning, knowledge of the subject matter, assessing student performance, and inadequate guidance and support. Given the problems teaching consumes, researchers have argued that teachers need support during their first years of teaching to increase their PCK [85, 86]. Liberman et al. [81] recommended that allowing CS teachers to emulate a lab session and understand the pressure of students in this context, might improve their understanding of the content knowledge and encourage them to develop an initial PCK.

Finally, effective teaching involves aligning the three major components of instruction: learning objectives, assessments, and instructional activities (Figure 3). It is important to engage learners' innate interests to further their retention, and to motivate their achievement.

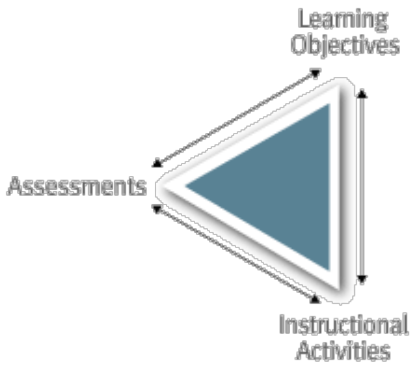


Figure 3. The three major components of instruction.

Teaching is more effective and student learning is enhanced when: the instructor articulates a clear set of learning objectives; the instructional activities support these learning objectives by providing goal-oriented practice; and the assessments provide opportunities for students to demonstrate and practice the knowledge and skills articulated in the objectives, and for instructors to offer targeted feedback that can guide further learning[9].

3.2 The Learning

In an attempt to design educational teaching and learning content, one has to be familiar with the most influential educational and learning theories of our times. Although there is no one definition of learning that is universally accepted by theorists and researchers, learning represents how individuals understand new things [87]. Learning theories provide principal techniques that should be used to improve the quest of learning through appropriate learning techniques according to the learner's need. Moreover, a theory provides people a description to add up of complicated methods and phenomena. There are many broadly-considered learning theories which influence at the most general level.

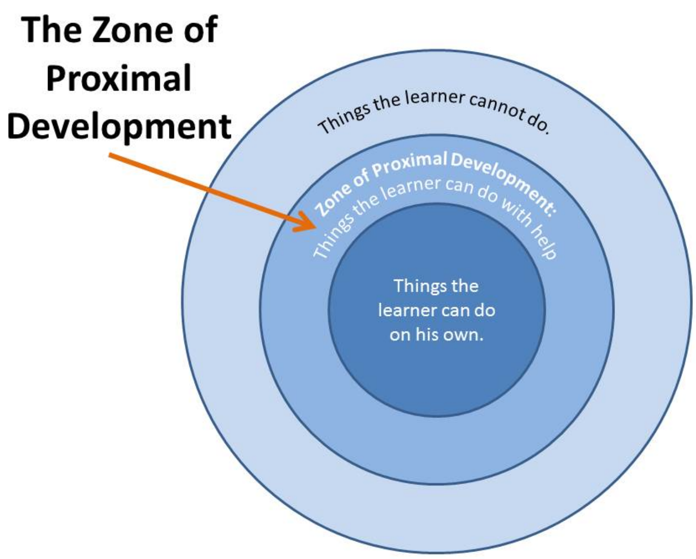


Figure 4. Vygotsky's ZPD learning theory.

[9] Eberly Center for Teaching Excellence & Educational Innovation, Carnegie Mellon University https://www.cmu.edu/teaching/principles/teaching.html, last accessed December 2016

For example, the work of Jerome Bruner which builds upon the structured stages of cognitive development outlined by Jean Piaget [88], emphasizes the relationship of cognitive structure to the structure of disciplinary content: "*What are the implications of emphasizing the structure of a subject, be in mathematics or history - emphasizing it in a way that seeks to give a student as quickly as possible a sense of the fundamental ideas of a discipline?*" [89]. Together with Bruner's work is the work of Lev Vygotsky. Vygotsky's ideas placed on the notion that knowledge and learning are culturally and socially constructed. Especially, his idea of the "*zone of proximal development*" (ZPD) has been considered very important (Figure 4).

ZPD states that students have limitations for progress they can make from their current knowledge state, but, with the help of a teacher giving appropriate interventions and scaffolding, their understanding can expand further than it would if they were left alone. "*…the distance between the actual level of development as determined by independent problem solving without guided instruction and the level of potential development as determined by problem solving under adult guidance or in collaboration with more capable peers*". [90]

Another collection of approaches, known broadly as behaviorist, grow from the work of B.F. Skinner [91] and focus only on objectively observable behaviors, therefore discounting internal mental activities. In a behaviorist environment, learning is considered to be the acquisition of new behavior. "*Conditioning*" for teacher-approval, high marks, or other reward, is the process by which learning occurs. Although they find some use in the classroom, these ideas are more often seen incorporated into various on-line pedagogic environments [92].

However, contemporary approaches introduce the concept of constructivism. What happens in teaching and learning is that learners actively construct knowledge, rather than stand as passive recipients [93]. This can lead directly to classroom implications, involving specific practices such as "*reciprocal teaching*" and "*jigsaw instruction*" [94, 95] whereby students learn through constructing their knowledge in order to teach others. In cognitive constructivism, whose principal proponent is Piaget, an individual's reactions to experiences lead (or fail to lead) to learning [93]. In addition, constructivist theorists, have extended the traditional focus on individual learning to address collaborative and social dimensions of learning with social constructivism. In social constructivism, which originated primarily from the work of Vygotsky, language and interactions with others (e.g. peers, teachers, family), take a primary role in the construction of knowledge [96]. There are also more general approaches about constructivism in CSE, for example Ben-Ari's work [97]. However, the most influential theory in the field of CS is the work of Seymour Papert, co-founder of the MIT Media Laboratory and collaborator of Jean Piaget [98]. His theoretical approach, called constructionism, is inspired by the constructivist theory that individual learners construct mental models in order to understand the world around them [99]. Constructionism advocates student-centered, discovery learning where students use information they already know to acquire more knowledge.

According to Ben-Ari [100], "*constructivist practices in CSE place the expectations on students to discover knowledge by themselves when placed in the appropriate situation*". He found that knowledge is acquired recursively: sensory data is combined with existing knowledge to create new cognitive structures, which are in turn the basis for further construction. Knowledge is also created cognitively by reflecting on existing knowledge. Furthermore, constructivists consider knowledge as a process that is created by the activity of the learner. This may seem contrary to traditional styles of instruction such as lecturing and reading books, but is in line with the constructivism paradigm [101]. Moreover, research in CSE [100] has shown that constructivist teachers actively guide students to construct "*viable mental models*".

In order to develop CT therefore, it is important to provide relevant motivations in the learning environment. One of the most popular instructional approaches which are developed from this focuses around the use of LEGO®[10] in education, although there are other constructionist approaches which do not rely on proprietary manipulables. More recently, the work of Lave and Wenger [102, 103] has extended the notion of the social nature of learning with ideas that learning is constantly situated within authentic situations, and takes place within groups of practice. Herewith, larger notions of how communities are structured and how learning occurs in them are associated. Learning and membership of community are closely identified with each other and knowledge cannot be separated from practice. This theory is valued for CS, a discipline that has a clear set of vocational directions, and it is one which has begun to be thoughtfully explored within the CS classroom [104].

Aside from the learning theories, there are also learning models and taxonomies. Learning taxonomies can be seen as a medium that can be used in a variety of educational contexts. Learning taxonomies, particularly Bloom's taxonomy of the cognitive domain [105], have had a considerable impact on curriculum and assessment design in the last fifty years. The cognitive domain, describes a range of cognitive behaviors found in educational assessment. The taxonomy comprises six levels, arranged along a continuum of complexity as shown in Figure 5 together with the revised version [106]. It was created in order to describe educational objectives that went beyond simple recall of fact and was aimed at "*teachers, administrators, professional specialists and research workers*" and was "*especially intended to help them discuss these problems of educational objectives with greater precision*" [105]. This influential educational taxonomy, has been widely used within CSE research over the last years [107-109].

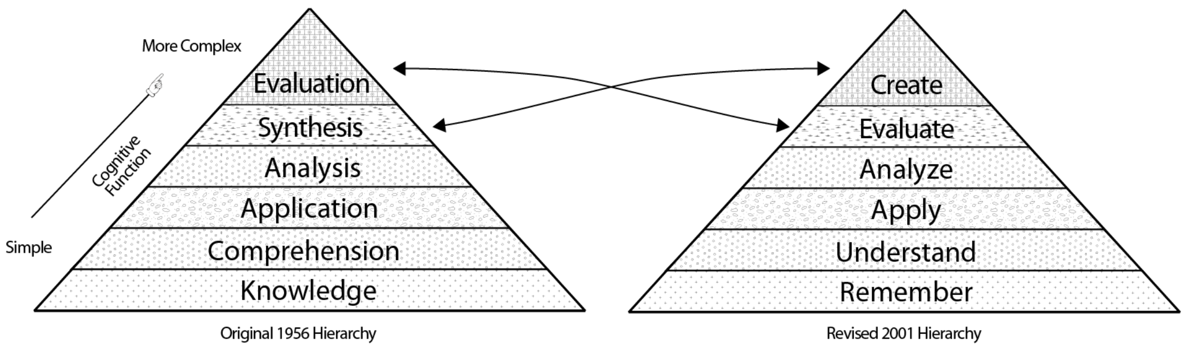


Figure 5. Bloom's Taxonomy Hierarchy (original and revised).

Another influential work, is Kolb's Learning Cycle [110], which aims to structure instruction so that experience is seen as the source of learning and development. As shown at Figure 6, in Kolb's cycle the process of learning seems to begin when a new experience is encountered. He argues that all points in the cycle act as equal partners in the learning process. The result is that the learning process can start at any one of the four points in the cycle. The original learning cycle presented a simplistic representation of this process. Recent versions (Figure 6) of the cycle have been changed and acknowledge the complexity of the learning process.

In contrast to these cognitive models, William Perry relates learning to a model of intellectual development maturity [111, 112]. He claims that students learn through nine levels with respect to

[10] LEGO® Education https://education.lego.com , last accessed January 2017

intellectual and moral development. These nine levels are grouped into four stages: dualism, multiplicity, relativism, commitment (Figure 7). These stages can be characterized in terms of the student's attitudes toward knowledge.

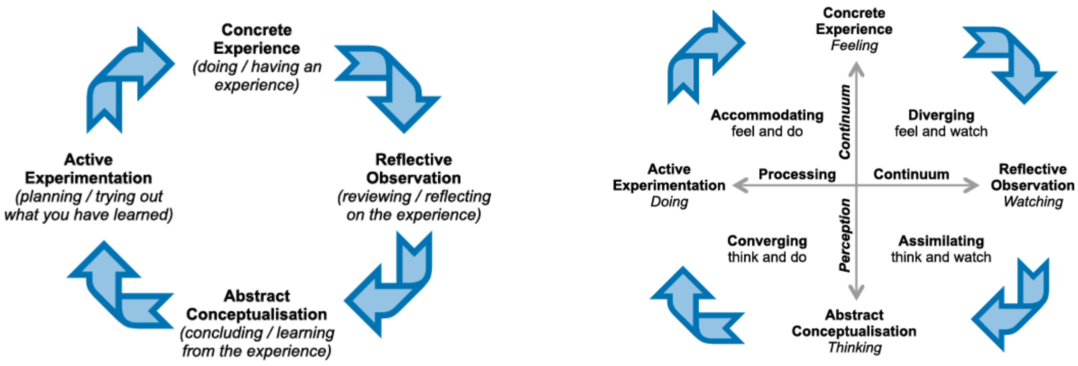


Figure 6. The four stages of Kolb's Learning Cycle (original and updated).

This learning process, or as called "*journey*" is sometimes repeated, and one can be at different stages at the same time with respect to different subjects. The aspect of learning that they have chosen, what they have chosen to emphasize in the creation of their model, what to simplify, and what to discard-all contribute to the differences between these constructs. Nevertheless, Bloom, Kolb and Perry have all devised models. Models have also been developed with regard to situated classroom instruction. One of the most lasting in recent times has been the model of Problem Based Learning which has spread widely in CSE (explained in 3.4.2).

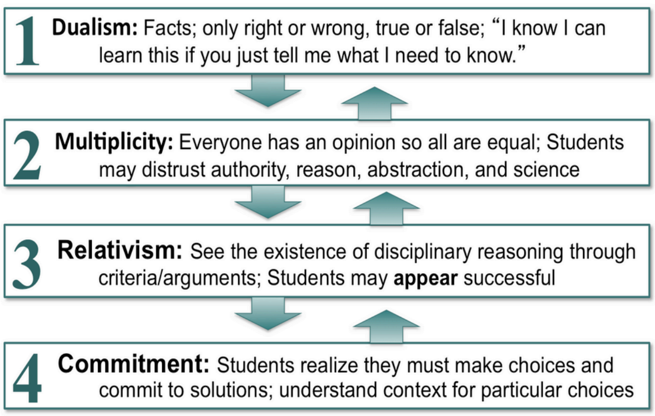


Figure 7. Perry's Intellectual Development Stages.

In addition, there are specific instruments devised to expose particular aspects of the instructional situation. Mostly these are manifest as questionnaires or scales, which describe students learning. For example, there are inventories on learning styles [110], on approaches to studying [113] and on personality types [114]. Their use within the classroom is predicated by the instructor being interested in a specific area, for example, in adjusting materials to groups of student with differing learning styles, or for assuring well-formed groups for a work. There have been attempts to organize the different theories of individual learning. Since most research moves around learning and cognitive style, the taxonomies primarily focus on the connections of the two constructs schemes [115]. A popular framework for the study of the different learning theories is Curry's onion model

[116]. Curry used the metaphor of an onion to describe the three levels of learning (Figure 8). He described the effect of personality traits as the inner layer of learning, information-processing styles as the middle layer of learning, and environmental and instructional preferences as the outer layer of learning.

The outer layer is the easiest to observe but it is also easily influenced by external factors and therefore, not very stable [117]. The middle layer of the onion model is the way the learner processes information and is more stable than the first, since it does not interact directly with the environment [110]. The inner layer is a relatively permanent and stable level since it is related to personality traits and deeper cognitive processes. It is the layer where the cognitive style dimensions [118, 119] can be placed.

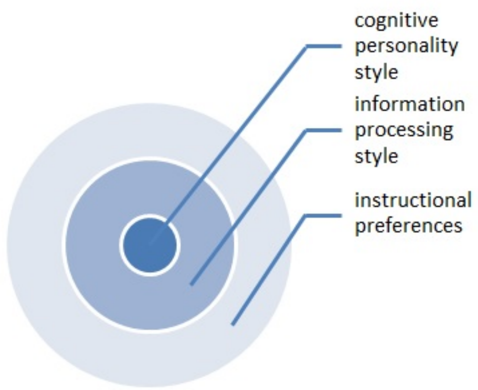


Figure 8. Curry's Onion Model.

Learning is a complex field, and modern research in the area should be conducted taking into account domains like psychology, sociology, linguistics, philosophy, etc.

3.3 Traditional Strategies

The roots of contemporary teaching and learning theories extend far into the past. From a philosophical perspective, learning can be discussed under the heading of epistemology, which refers to methods of acquiring knowledge. There are two main theories about knowledge and its relationship to the environment: rationalism and empiricism. Rationalism holds that the most important element for our acquisition of knowledge is reason and not experience (empiricism). The traditional theory of empiricism, which is dominant for centuries in education, considers the existence of an absolute knowledge, talking about objective reality which exists independently of human perception and that learners just absorb. This traditional strategy is relied on the idea that experience is the only source of knowledge [93]. The role of the teacher in traditional teaching is through lectures to impart the knowledge to his students and finally to evaluate their abilities [120]. Students are not encouraged to make their own interpretations. The teacher explains the facts using teaching strategies that control and manipulate the learning process. However, this transmission of knowledge assumes that everyone understands a concept by the same way. The reality is unique and it is structured and that structure can be presented to students. Empiricism claims that the mind is simply aimed to reflect this reality and its structure. The point made in this mental process is external to the learner and is determined by the structure of the real world [120, 121].

On one hand, educators based on the content of the courses taught prefer the traditional passive methods of teaching sometimes. Methods like lecture-based instruction has been shown to be most effective when the goal is to transmit information, and when organization and clarity are desired [122]. Charlton [123] suggested that lectures are an effective teaching method because they exploit human psychology and evolved nature to improve learning. He argued that, in many circumstances and for many students, lecturing is probably the best teaching method, especially for communicating conceptual knowledge and where there is a significant knowledge gap between lecturer and audience.

On the other hand, traditional teaching lacks on several points. Probably the main argument posed against traditional classroom styles involves how little it truly engages students [124]. The simple lecture based teaching definitely has its place in education, since there are subjects and concepts beneficial to traditional talking structures, and therefore there is no reason to completely abandon them. However, enhancing lectures with something more interactive might be all a teacher really needs to get students learning and retaining knowledge [125].

Moreover, Kolb [126] identifies that different learning styles do not benefit the same from the traditional lecture structure. Aligned, Fleming and Mills [127] recognize at least four general learning styles: Visual, Aural, Read/write, and Kinesthetic (VARK). Fleming and Baume [128] found that only visual/verbal and auditory/verbal learners get anything out of lectures, so tactile/kinesthetic and visual/nonverbal students end up falling behind. Educators might not completely reach all styles and might want to explore different ways to infuse other activities and projects into the syllabus so everyone gets a chance to see lessons in a way that makes sense [129]. Traditional, lecture-based structures serve particularly adroit conduits for rote learning and memorization [124]. Essential components to a well-rounded education, to be certain, but ones often over-emphasized in contemporary classrooms.

Another issue with relying largely on traditional strategies involves teacher bias. Most subjects are not objective, and since the instructor stands as the highest authority in the room, the more strict, rote structure only presents perspectives on the matters at hand [130]. The most effective educational settings are more active and allow students to consider content from multiple angles and form multiple opinions, rather than constantly repeat what teachers transfer [124].

Furthermore, traditional teaching is not the right fit for every subject. STEM related fields benefit from more interactive, hands-on approaches [131]. In addition, a hybrid system engaging multiple learning styles and ideologies encourages more critical thinking and analysis skills than merely listening [126]. Many schools and teachers consider traditional teaching and learning as a one-size-fits-all approach. Unfortunately, they do not always consider whether it works with the content at hand. Just listening the lesson works best for some subjects, not all, and its main drawback involves compromising critical thinking skills [132].

Moreover, not every teacher is good at public speaking. Poor communicators and speaking anxiety can seriously screw over different learners, even if they typically benefit from traditional lecture structures [133]. When it comes to public speaking, it is crucial that teachers understand where their limitations lie and alter their styles accordingly for maximum educational achievement [134].

Also, the amount of information retained by students, declines substantially after ten minutes [135][11]. Another study suggests that this number might hover between 15 and 25 minutes [136].

[11] http://news.bbc.co.uk/2/hi/uk_news/education/8449307.stm, last accessed January 2017

Nevertheless, such a limited attention span benefits from the very same variances that promote enhanced engagement. They provide all the education of a traditional lecture where students are passive recipients of information transmitted by the teacher and they are highly dependent on her for much of their learning needs. They mainly develop memorization skills since they recall and repeat facts. In addition, there is no real way to find out whether students understand the subject until assignment or test time comes. Students acquire unmotivated knowledge that is often difficult to apply in the real environments, whereas "*What matters is not just what students know but what they can do with what they know. What is at stake is the capacity to perform, to put what one knows into practice*" [137].

Perhaps expectedly, the whole thing about traditional strategies is that they that only suit better some students than others and also that they strengthen a limited number of skills. As a passive format, problem-solving, creativity, collaboration, critical thinking, analysis, and other more active elements of a 21st century education receive little attention [138]. Traditional methods certainly support memorization and note taking, but they are not the only abilities students need to succeed. Therefore, reliance on the traditional lecture as the main mode of student learning has been criticized and new alternative teaching methods are needed.

3.4 Alternative Methods

In addition to traditional teaching and specific program content, educators and curriculum designers must consider learning activities and instructional techniques that aim to student motivation and learning. As Morrison et al. [139] state, instructors at any level of education should implement various pedagogical approaches and teaching methods when the discipline and learning tasks vary. There are also many other parameters that need to be considered, like the students' age and experience on technology matters or the learning goals. Alternative methodologies can provide the opportunity to deal with the complexity of each formal or informal class.

Alternative is considered everything that is relating to activities that depart from or challenge traditional norms[12]. Alternative Teaching Methods (ATM) that support students' active involvement in the learning process, according to the constructivism and constructionism principles, could be very helpful. There are several methods used by educators and researchers in order to deal with the complexity and the needs of various cases. The use of them though could be adapted in more approaches with similar features. The following sub-sections describe every known teaching method, strategy and approach that can be applied within CSE.

3.4.1 Peer Learning

Peer Learning (PL) method has a long history. It has always taken place directly or indirectly and is as old as any form of collaborative or community action [140]. PL refers to students learning with and from each other as fellow learners without any implied authority to any individual. It is based on the principle that "*Students learn a great deal by explaining their ideas to others and by participating in activities in which they can learn from their peers*" [141]. PL can be defined as the acquisition of knowledge and skills through supporting and active helping among status equals or peer matched companions [140]. It involves people from similar groups who are not professional educators helping each other to learn and learning themselves.

[12] Cambridge Online Dictionary: http://dictionary.cambridge.org/dictionary/english/alternative , last accessed December 2016

PL is a well-established method in many disciplines especially in science [142], technology, engineering, and in studies around STEM subjects [143, 144]. Research in computing education has shown that the discussions held between peers positively contribute to their engagement [145] and their understanding [146, 147]. Students also achieve higher grades with PL comparing to traditional teaching [148, 149]. Porter et al. [146] researched another indicator of student success, which is the rate at which students pass the course or, conversely, the rate at which they fail. They concluded a 10-year study in several CS courses at university level and they found that adoption of PL in the classroom reduces fail rate significantly compared to traditional instruction. Moreover, there is evidence that PL increases self-efficacy and is enjoyed by students and instructors alike [149, 150].

The PL methodology encompasses a broad number of activities. For example, Griffiths et al. [151] identified 10 different models of PL. These ranged from the traditional proctor model, in which senior students tutor junior students, to the more innovative learning cells, in which students in the same year form partnerships to assist each other with both course content and personal concerns. Other models involved discussion seminars, private study groups, counseling, peer-assessment schemes, collaborative project or laboratory work, projects in different sized (cascading) groups, workplace mentoring and community activities. However, the most intensively researched forms of PL are Peer Tutoring (PT) or else called Peer Teaching, Cooperative Learning (CL) and Peer Assessment (PA). They have all been researched more in school education than in other contexts.

PT is characterized by specific role-taking as tutor, with high focus on curriculum content and usually also on clear procedures for interaction, in which participants receive generic and/or specific training [152]. Some PT methods scaffold the interaction with structured materials, while others prescribe structured interactive behaviors that can be effectively applied to any materials of interest. It is a more instrumental strategy in which advanced students, or those in later years, take on a limited instructional role. It often requires some form of reward for the person acting as the teacher. PT is a well-established practice in many universities, whereas common PL is often considered to be related a component of other strategies, such as the discussion group [153].

CL is an education strategy that uses student peer groups as orchestrated learning environments [154]. Corporation is more about working together and it has been described as "*structuring positive interdependence*" [155] in pursuit of a specific shared goal or output. This is likely to involve the specification of goals, tasks, resources, roles, and rewards by the teacher, who facilitates or more firmly guides the interactive process. Typically operated in small groups of about six heterogeneous learners, CL often requires previous training to ensure equal participation and simultaneous interaction, synergy, and added value. Group work has long been an important component of CSE since it models the way software development is done in industry. However, CL involves manipulating groups into planned cooperative learning environments which is relatively new and uncommon in the CS discipline [156].

In addition, Mazur [157] developed his version of PL called Peer Instruction (PI). He adjusted the PL approach to ensure students could understand and apply core concepts to problems rather than just use memory recall and improved his students' performance significantly. In his method the educator uses qualitative (usually multiple choice) questions that are carefully constructed to engage student difficulties with fundamental concepts. Students consider the problem on their own and contribute their answers in a way that the fraction of the class giving each answer can be determined and reported. Then students discuss the issue with their classmates for a couple of minutes and answer again. Possible issues are resolved with a class discussion and clarification.

Finally, PA or self-assessment involves students taking responsibility for assessing the work of their peers against set assessment criteria. It is a process whereby students or their peers grade assignments or tests based on a teacher's benchmarks [158]. Boud [141] suggests that ways of assessing its value must be explored together with strategies for its effective implementation bearing in mind the following features:

- Key learning outcomes. Assessment needs to focus on the desired outcomes and therefore, clarity needs to be sought about the outcomes.
- Holistic design. Ensure that assessment tasks purposefully include, not marginalize, PL.
- Consequence. In designing assessment, ask: "How can assessment activities support meaningful engagement for students?"
- Lifelong learning. The range of assessment tasks should leave students better equipped to engage in continuous learning. That is, those that encourage working with others, planning, organizing knowledge. working interdependently.
- Reflexive activities. PL activities have an advantage over other teaching and learning strategies in that they have considerable potential to promote critical reflection, provided the climate of reciprocal communication and openness is encouraged.

It is important to recognize that PL is not a single practice. It covers a wide range of different activities each of which can be combined with others in different ways to suit the needs of a particular course.

3.4.2 Problem Based Learning

The Problem-Based Learning (Problem-BL) method was developed for medical education since the 1960' s and since then it has been broadened in several learning settings and disciplines like CS. Problem-BL is "*based on the principle of using problems as a starting point for the acquisition and integration of new knowledge*" [159]. In this teaching approach learners use "*triggers*" from a problem case or a scenario to define their own learning objectives [160]. Therefore, students learn about a subject through the experience of solving a problem which does not focus on with a defined solution, rather solving a large, real-world problem [161]. Problem-BL is a student-centered pedagogy where lectures are replaced by additional tutorials and laboratory time and appropriate problems are used in order to increase knowledge and understanding. The process Problem-BL method is clearly defined, and learners follow a series of steps to find the solution into several variations of problems. It can be thought of as a collaboration teaching method that combines the acquisition of knowledge with the development of generic skills and attitudes [162].

The effectiveness of Problem-BL has been studied over the years. It enhances problem-solving skills and allows the development of many other attributes including knowledge acquisition critical appraisal, communication, collaboration, teamwork, independent responsibility for learning, sharing information, and respect for others [160, 163]. Moreover, it enhances active learning and therefore better understanding and retention of knowledge [162]. Problem-BL is an effective way of used in many educational settings and offers several advantages over traditional teaching methods, including motivating the students, encouraging them to set their own learning goals, and giving them a role in decisions that affect their own learning [164]. It also empowers learners to conduct research, integrate theory and practice, and apply new knowledge to find solutions to problems [165]. Finally, through Problem-BL learners develop life skills that are applicable to many domains [166].

There are various specific ways for implementing the Problem-BL method. Schmidt [167] presented a process of seven steps to use the method effectively (Table 3).

Table 3. Steps involved in PBL.

Step 1:	Clarify terms & concepts
Step 2:	Define the problem
Step 3:	Analyze the problem
Step 4:	Restructuring and inventorying
Step 5:	Formulate learning objectives
Step 6:	Gather information from outside the group
Step 7:	Synthesize and test the new acquired information

The aims of this method is to foster learning by:

1. activating the prior knowledge of students about the topics to learn,
2. connecting the learning to specific problem situations that might occur in practice, and
3. making the students to elaborate the material that they have learned.

Problem-BL is generally introduced in the context of a defined core curriculum in education. The method has also been suggested as a solution to CSE, and have been implemented in several cases. For example, the use of Problem-BL in 1st year undergraduate CS courses discarded lectures in favor of learning in the context of solving problems with a 'real life' flavor and emphasized generic skills such as group work, independence and initiative, planning, and problem-solving [164]. O'Kelly and Gibson [168] in their study combined elements of fun, programming, games, AI and competition through a Problem-BL approach. They encouraged their students with the fun element of creative ideas within the constraints of the RoboCode environment with the challenges of refining these ideas into a workable solution. The authors argue that the problem transformed fragile knowledge into a concrete transferable skill that can be applied in new situations while students developed skills for each stage of the software development process: requirements analysis, design, implementation, and testing; and thought critically, reflected on their work, conducted tradeoffs and make informed decisions.

However, one of the major difficulties to the implementation of Problem-BL is the lack of a good set of problems. Duch identified five characteristics of what makes a PBL problem good [169]:

1. Effective problems should engage the students' interest and motivate them to probe for deeper understanding of the concepts being introduced.
2. PBL problems should be designed with multiple stages.
3. The problems should be complex enough that cooperation from all members of the group will be necessary in order for them to effectively work towards a solution.
4. The problem should be open-ended.
5. The content objectives of the course should be incorporated into the problems.

Within PBL the focus is shifted from teaching to learning and this shift in conjunction with a good problem provides each student with the freedom to think for themselves, activate their prior knowledge and acquire new knowledge in an explorative and creative way.

3.4.3 Project-Based Learning

Project-Based Learning (Project-BL) has its roots back in the work of Dewey and dates back to 1918 when the term was first used[13][14]. Project-BL is often confused with Problem-BL since it is a teaching method where students learn by investigating a complex question, problem or challenge. However, in Project-BL students explore real-world meaningful problems that are important to them and find answers through the completion of a project. Students also have some control over the project they will be working on, how the project will finish, as well as the end product. A project-based classroom allows students to investigate questions, propose hypotheses and explanations, discuss their ideas, challenge the ideas of others, and try out new ideas.

Project-BL is an overall approach to the design of learning environments. The Project-BL environments comprise the following five key features [170, 171]:

1. Teachers pose a driving question, which is the problem to be solved.
2. Students explore the question by participating in authentic, situated inquiry - processes of problem solving. Through that process, they learn and apply important ideas in the discipline.
3. Students and teachers engage in collaborative activities to find solutions to the driving question. This mirrors the complex social situation of expert problem solving.
4. Students are scaffolding with learning technologies that help them participate in activities normally beyond their ability.
5. Students create a set of tangible products that address the driving question. These are shared artifacts, publicly accessible external representations of the class's learning.

Research has demonstrated that students in project-based classrooms achieve higher scores than students in traditional settings [172-174]. After completing a project, students understand content more deeply, remember what they learn and retain it longer than is often the case with traditional instruction. Students who gain content knowledge with Project-BL are better able to apply what they know and can do to new situations. Project-BL engages students and promotes active learning, and provides real-world relevance for learning which allows the development of higher order thinking skills [175]. In addition, Project-BL builds success skills for college, career, and life all essential for the 21st century which requires more than basic knowledge and skills [176]. In a project, students learn how to take initiative and responsibility, build their confidence, solve problems, work in teams, communicate ideas, and manage themselves more effectively [170]. Furthermore, Project-BL provides opportunities for students to use technology effectively [171]. Students nowadays are familiar with a variety of tech tools that are a perfect fit with the method. Finally, projects allow teachers to rediscover the joy of learning alongside their students by working more closely with active, engaged students doing high-quality work. Therefore, Project-BL makes teaching more enjoyable and rewarding for all participants.

[13] Buck Institute for Education: http://www.bie.org/, last accessed December 2016
[14] Edutopia: https://www.edutopia.org/blog/pbl-vs-pbl-vs-xbl-john-larmer, last accessed December 2016

Both Problem-BL and Project-BL are referred to as PBL, and some find it confusing to separate the two pedagogies. They have several common aspects:

- involve realistic problems and situations,
- are based on authentic educational goals,
- include formative and summative evaluation,
- are students centered and teacher facilitated,
- are intrinsically engaging and motivating,
- are multidisciplinary,
- improve research and problem-solving skills, as well collaboration with their peers.

The difference between the two teaching methods is that students who complete Problem-BL often share the outcomes and jointly set the learning goals and outcomes with the teacher. On the other hand, Project-BL is an approach where the goals are set. It is also quite structured in the way that the teaching occurs. In addition, project-based follows general steps while problem-based provides specific steps. Essentially, Project-BL often involves authentic tasks that solve real-world problems while Problem-BL uses scenarios and cases that are perhaps less related to real life [177]. Perrenet et al. [178] made a comparison of the two methods. They noted that the two approaches differentiate at the following:

- Project tasks are closer to professional reality and therefore take a longer period of time than problems.
- Project work is more directed to the application of knowledge, whereas Problem-BL is more directed to the acquisition of knowledge.
- Project-BL is usually accompanied by subject courses, whereas Problem-BL is not.
- In Project-BL, management of time and resources by the students as well as task and role differentiation are very important.
- Self-direction is stronger in project work, compared with Problem-BL, since the learning process is less directed by the problem.

In conclusion, both teaching methods have their place in today's classroom and can promote 21st century learning. It is more the importance of conducting active learning with students that is worthy and not the actual name of the task.

3.4.4 Studio-Based Learning

Studio-Based Learning (SBL) is a teaching method that emphasizes in collaborative and design-oriented learning [179]. This pedagogy approach has its roots extend as far back as Platonism and it is fully realized by architectural schools and industrial design education [180]. The SBL model is in the form of the "*design studio*", a place where students set up their own works paces, and create and present their designs [181]. As students work on design tasks, they develop a "*community of practice*" providing support and feedback to each other. SBL involves a series of design problems, which may either be a sequence of progressively more challenging design problems, or various components of a large design project. A key aspect of the design studio is the design critique. Design critiques are review sessions in which students present their evolving solutions to the instructor and the class for feedback and discussion [182].

From the literature [183, 184] the key characteristics of the SBL method are:

a. Classroom assignments should primarily be project-based. Students are given meaningful problems for which they have to design and implement computational solutions individually or in groups.
b. These problems are open to multiple solution strategies. This means that students have to consider alternate solutions in terms of efficiency and software engineering considerations, choose the best, and justify their choice.
c. They must then articulate their solutions and justifications to the entire class for peer review, feedback and discussion.
d. Their peers and the course evaluate these and provide comments and criticisms.
e. Students are given the opportunity to respond to this feedback and modify their solutions appropriately.

Furthermore the basic learning activities that define the SBL approach in CSE are "*representation construction*" and "*presentation and discussion*" [179]. Students construct their own visual and verbal representations of the computing concepts, algorithms, or processes under study; and then in design critiques, they present their representations for discussion and feedback from their peers as well as the instructor.

Through the SBL approach there are several benefits for students. There are not only improved learning outcomes [179], but also to a stronger sense of community, improved critical thinking skills, and prevention of premature convergence to incomplete knowledge [185, 186]. Students get experience in individually and collaboratively solving algorithm and software design problems [183]. They also are more engaged, invested and motivated in classrooms employing the SBL pedagogical model [187]. In addition, the methods enhances presentation skills through oral presentations and argumentations [180]. Finally, students increase their enjoyment, motivation and interest in CSE [179, 188], and become more proficient practitioners of computational problem-solving by getting experience on learning from the design exercises [183, 184].

In many CS courses, classroom assignments are already project-based. Researchers have noted numerous similarities between CS and design sciences like as architecture and have implemented the SBL model in CSE [188, 189]. According to Hundhausen et al. [179] the SBL method has three key features that make it particularly well-suited to the varying needs of CSE:

i. *Scalability*. The approach scales well to different computing courses and class sizes. Any computing course includes concepts and processes that learners can represent, present, and discuss. Moreover, these activities can be implemented in weekly studios that replace labs.
ii. *Adaptability*. The approach can be easily adapted to meet the local needs of specific educational settings and instructors. Rather than replacing traditional teaching, the core activities SBL are intended to complement the lecturing.
iii. *Technology independence*. While implementations of the SBL will clearly benefit from visualization or concept representation software, the approach is designed to be completely independent of any specific technology. Instructors can choose to have students develop

their representations using a variety of technologies and make use of specific technologies with which they are familiar and comfortable.

Furthermore, Hundhausen et al. [190] developed their active learning method based on STL, called the Pedagogical Code Review (PCR). PCR is a collaborative activity in which a small team of students, led by a trained moderator: (a) walk through segments of each other's programming solutions, (b) check the code against a list of best coding practices, and (c) discuss and log issues that arise. Results of using PCR provide evidence that PCR can promote positive attitudinal shifts, and hone skills in critical review, teamwork, and communication [190, 191].

3.4.5 Inquiry-Based Learning

Inquiry-Based Learning (IBL) is an approach in which students are actively engaged in the learning process by asking questions, interacting with the real world, and devising multiple methods to address the questions [192, 193]. IBL is a pedagogical method developed during the discovery learning movement of the 1960s as a response to traditional teaching [194]. It includes Problem-BL, and is generally used in small scale investigations and projects [195]. Learners in IBL are guided by questions that lead to gathering of evidence, formulating explanations from the evidence, communicate, and justify the explanations. The teacher plays the role of a cognitive guide and a facilitator in the process. There is a range of IBL designs or levels that vary in whether the problem, procedure and solution are given to, or constructed by, the learner [196]. A well-established practice in CS is the guided inquiry (described in section 3.4.6), in which the decision-making for the course is shared between the instructor and the learners.

IBL shares common features with the theoretical framework of the Learning Cycle (Figure 9): exploration, concept development and application [197].

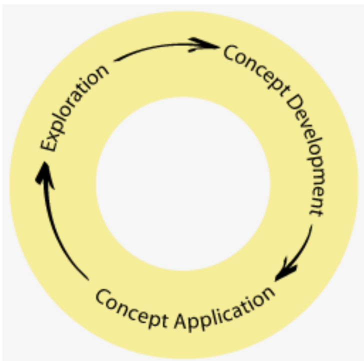


Figure 9. Diagram of the Learning Cycle [197].

This three phased theory (Figure 9) of the Learning Cycle by Atkin and Karplus [198], aimed to rationalize how students learn. The three core phases: Exploration is the inquiry phase where students deal with new phenomena at the concrete level of reasoning; Concept Development is where concepts are introduced which explain the phenomena of the previous phase, and; Concept Application is the phase where students perform experiments which apply new acquired knowledge.

CS is well-suited for an inquiry-driven educational approach since the discipline itself is changing so rapidly that it is difficult to introduce students to it without involving them in the creative process [199]. Moreover, computing technologies and their fundamental properties offer dramatic, new opportunities to support the IBL method [170].

Perhaps the most noted benefit of the IBL learning in CS is that learners take active role in their education [199]. It also provides them with opportunities to engage in complex tasks that otherwise would be beyond their current abilities [200]. According to Hmelo-Silver et al. [200] IBL is a powerful and effective technique, where learners can access difficult tasks more easily through the scaffolding learning process and can manage complex assignments within the ZPD (Figure 4).

3.4.6 Process Oriented Guided Inquiry Learning

Process Oriented Guided Inquiry Learning (POGIL) shares characteristics with active and discovery learning and combines the effective learning practices of CL and IBL methods. POGIL is based on learning science and was originally developed in college chemistry courses in the mid 1990's [201]. In this teaching method, learners work on teams and the instructor gives scripted inquiry activities and investigations to help them construct their own knowledge [202]. The teams follow processes with specific roles, steps, and reports that help the members to develop process skills and encourage individual responsibility and meta-cognition. Active facilitation is a key aspect of POGIL and the instructor is not a lecturer or passive observer.

However, Brown [203] notes that "*implementation of POGIL into a course is not an effortless proposition*" as it is so different from traditional classroom situations. While there is no single way to incorporate POGIL in the teaching and learning process, a typical implementation will involve the following 3 phases:

- First, students in small groups (3 to 5 learners) with each student assigned a unique role (e.g. manager, spokerperson) explore models or data and generate hypotheses to help understand or explain what they observe.
- Second, the patterns or hypotheses are used to define or invent a new concept; importantly, students have constructed understanding before the concept is introduced. At this phase are used the materials that guide higher-level thinking and learning.
- Finally, the new concept is applied in other situations or contexts to help students generalize its meaning and applicability. Thus, the main characteristic here is the shift in the role of the instructor from disseminator of information to facilitator of the learning process. Rather than always serving as the direct source of information, the instructor continually assesses how and when to offer additional guidance as the students work together to construct knowledge of course content [204].

In POGIL method all students fully participate in the learning process. A successful POGIL activity guides them through a series of questions to construct an understanding of the material for themselves, rather than providing an explanation of the material in lecture or written format.

More recently, POGIL has been introduced in biology, engineering, mathematics, and STEM disciplines. Several CS classes incorporate the pedagogy, including CS1[15], CS2, software engineering, and even more specified classes like neural networks and fuzzy systems [205-209]. Much of this widespread expansion can be attributed to the POGIL Project[16], which seeks to provide structured support for the dissemination, implementation, and assessment of student-centered pedagogies like POGIL [210]. POGIL is an evidence-based approach, and has been shown to significantly improve

[15] CS1 and CS2 commonly refer to introduction to programming courses and basic data structures courses respectively.
[16] The POGIL Project: https://pogil.org/, last accessed January 2017.

student performance [209, 211, 212]. It seems easier for the students to learn the programming concepts by following POGIL [213]. Hu and Shepherd [204] successfully implemented the method in a scientific computing course to teach science students how to program in Python. The authors have developed guided inquiry activities[17] that lead learners to discover and understand even the more difficult programming concepts like recursion and provided student evaluations and an assessment on how good students learned to program comparing to traditional CS labs and other active-learning pedagogies. Structured roles used in POGIL, help students learn to communicate and work in teams more effectively and develop skills such as communication, teamwork, critical thinking, and problem solving [214]. Yet, the method is beneficial for instructors too. Hu et al. [215] conducted an analysis of CS faculty perceptions of POGIL. Participants strongly agreed that POGIL method helped them overcome several problems during their teaching since students were more engaged and active. Although the preparation time was an issue, most participants reported better learning outcomes for their classes. Finally, research has shown that POGIL is relatively effective in incorporating voice and higher order thinking to better teach for diversity and attract students to take more CS courses in higher education [216].

Because of its success, the POGIL approach has been adopted by the CS community and has generated increasing interest and activity at the ACM's special interest group of CSE (SIGCSE) [209, 217]. The flagship SIGCSE conference, which has been held annually since 1970 with thousands of attendees, holds a permanent place for a POGIL workshop. During the POGIL special session[18], attendees experience several activities, and learn how the method can transform CS classes at all levels, from high school to graduate-level classes, from small schools to large universities [211]. POGIL is suitable for every CS context, level and course [218] and such engaging and inclusive pedagogies and practices that build students' problem solving, abstract thinking skills, as well as their self-confidence, are necessary to contribute to the success of every student [219].

3.4.7 Team Learning

Team learning (TL) refers to a wide variety of instructional methods and programs that educators use to describe the collaborative work in order to achieve a common goal within the group [220]. The aim of TL practices is to attain the objective through dialogue and discussion, conflicts and defensive routines, and practice within the group. It involves the interaction of students learning from each other as well as from the task at hand. The learning takes place through the transfer of skills by observing others in action, collective problem-solving and experimentation, questioning assumptions and reviewing outcomes as a group. Although TL focuses on the abilities of a group working together more closely to collaborative learning and is a different teaching method than corporative learning (see section 3.4.1). The underlying premise of these techniques is that learning is enhanced by peer interaction. While most of the teaching methods presented in this chapter involve students work in teams, this section aims to present specific approaches of collaboration within CSE.

3.4.7.1 Collaborative Learning

Collaborative learning is commonly when groups of students work together to search for understanding, meaning, or solutions to create an artifact or product of their learning [221]. The

[17] Resources to support the use of POGIL in Computer Science (CS) and related disciplines: http://cspogil.org, last accessed January 2017.
[18] SIGCSE 2017 Workshop: Guiding Students to Discover CS Concepts and Develop Process Skills Using POGIL http://sigcse2017.sigcse.org/attendees/workshops.html, last accessed January 2017.

method has some essential differences than Cooperative Learning (Table 4); mainly it requires the mutual engagement of all participants and a coordinated effort to solve the problem whereas Cooperative Learning requires individuals to take responsibility for a specific section and then coordinate their respective parts together [222].

Table 4. Differences between Cooperative and Collaborative Learning [222, 223].

Cooperative	Collaborative
Students receive training in small groups and develop social skills.	Students already have the necessary social skills, and they build on their existing skills in order to reach their goals.
Activities are structured with each student having a specific role.	Students organize and negotiate efforts by themselves.
The individual constructs learning.	Learning is co-constructed, challenged, modified, shared understanding.
The activity is monitored by the instructor. He observes, listens and intervenes when necessary.	The activity is not monitored by the instructor. He just guides the students to the information needed.
Students submit their work for evaluation at the end.	Students retain drafts to complete further work.
Focus more on individual achievement.	Focus more on group achievement.

Collaborative learning facilitates:

- cognitive constructivism, where peer discussion leads to improved conceptual understanding [224].
- teamwork and deeper understanding that are promoted among team members with varying levels of prior knowledge [225].
- creativity, by transforming knowledge into a product that is fulfilling for the student [226].
- communication skills and critical thinking, by public oral presentations [227].

3.4.7.2 Pair Programming

Collaborative learning involves grouping or pairing students to work together. Such a method in CS is Pair Programming (PP). PP is the very simple concept where two programmers work together at one computer on the same task [228]. In PP there are two roles that the participant learners take: one is the "*driver*" actively creating code and controlling the keyboard and mouse; the other is the "*navigator*" who constantly reviews the keyed data in order to identify tactical and strategic deficiencies, including erroneous syntax and logic, misspellings, and implementations that don't map to the design [229]. After a designated period of time, the partners reverse roles. Code produced by only one partner is discarded, or reviewed collaboratively before it is integrated. Finally, paired students are as able to apply their learning independently as students who work alone [230].

Although PP has been used in industry since the 1970s [231], it has become much more popular in education in the last 10 years as part of the agile software development movement [232]. PP in education is frequently used in introductory programming courses as there is notable evidence that its use helps students learn to program. However, it has also been used in school education, in upper division courses such as software engineering or object-oriented programming, and even at the graduate level [233, 234].

There are many significant research findings about the benefits of PP in educational settings. PP has been found to significantly outperform individual programmers in terms of program functionality and readability, to report greater satisfaction with the problem-solving process, to have greater confidence in their solutions, and to be more likely to complete a programming assignment [229, 230, 235]. Furthermore, paired teams produce higher quality code in about half the time when compared with solo programmers [236]. Learners in paired teams have very positive attitudes from their programming experiences [237, 238]. PP provides enjoyment to the participants and they have increased intent to further pursue degrees in CS [230].

Concluding, PP seems to be a very effective educational technique. There is evidence that the method is effective in the classroom, but much more research is needed [239]. There are no widespread studies, multi-institutional studies or longitudinal studies, and there are very few replicated studies. Little is also understood about how best to teach students to pair program.

3.4.7.3 Participatory Learning

Participatory teaching approach is a form of a reflective teaching which is sometimes termed as interactive or learner-centered teaching method. In general, participatory methods are those which draw the student into the learning process and thus student becomes a participant who articulates in some way what is learned [240]. Participatory methods expect a high degree of activity and personal involvement of participants in the learning process. They are designed for small groups of participants, but their advantage is that they encourage better retention of knowledge. Typically, participatory methods are considered brainstorming, role playing, workshops etc. However, this section intends to present the method of Participatory Design (PD) as it is used in CSE settings.

PD which is also known as co-operative design or co-design, is an approach to design attempting to actively involve all stakeholders in the design process to help ensure the result meets their needs and is usable [241]. Thereby, every stakeholder gets a chance to understand the others' views and to benefit from their expertise. The field of PD grew out of work beginning in the early 1970s in Norway, when computer professionals worked with members of the Iron and Metalworkers Union to enable the workers to have more influence on the design and introduction of computer systems into the workplace [242]. Early, this work took the form of experiments in universities and currently is used in education. In that view PD is considered as a set of theories, practices, and studies related to end users as full participants in activities leading to software and hardware computer products and computer based activities [243, 244]. User participation relates to the involvement of users in development activities with varying form and degree of involvement and influence [242].

The PD method is most used within the Human-Computer Interaction (HCI) field [245-250]. In HCI students must be familiar with techniques of software design that embrace the human activity as an integral component of the analysis, design, and evaluation. In the PD process, the designer engages in analysis activities that provide an insight into the user's conceptual model or mental model of the tasks for the system that is being targeted for development [251]. Weinberg and Stephen [246] researched PD by using the method several years at their university course. Students learnt not only to build technically good systems but also to take into account the perspectives of organizations and individual users in particular roles, discover their needs and requirements, and transform them into appropriate (user-oriented) designs. Hecht and Maass [250] reflected on how to teach methods used in PD to graduate students. They provide evidence that the concept has proved very successful both in terms of student satisfaction and course results. The authors also invite researchers to share their experiences and start a discussion about teaching with PD. Moreover, PD has been tested in settings other than university. Antoniou and Lepouras [249] combined elements of PD techniques with focus groups in order to involve young children in the design process of a game application for museums. Their method assisted elementary pupils in visualizing the possibilities of using games in museums and demonstrated their ability to generate concepts and inspire the development of new technologies.

Computing professionals have to be prepared to meet not only technical, but also organizational, social, and political challenges, and therefore contemporary education has to include approaches like user participation and PD [248]. Using participatory teaching methods demands that teachers are well prepared and the course is fully planned. Moreover, successful learning is not achieved only by teachers' efforts, but also by students and thus their active participation and interest in the learning process are essential elements for success.

3.4.8 Game-Based Learning

Game-Based Learning (GBL) refers to teaching and learning activities carried out in formal and/or informal educational settings, by adopting games. In GBL methods the instructor uses tips, techniques, and tools that apply the principles of game design to the learning process [252]. It is a dynamic way to engage learners and help instructors assess the learning. GBL involves the use of both games designed clearly for fulfilling learning objectives (educational games) and games that are not developed for education (but rather for fun) nonetheless are used to pursue learning objectives [253]. In GBL, games are used as a tool in order to provide learning and training through a pleasant experience [254-256].

The terms GBL and Digital Game-Based Learning (DGBL) describe learning through gaming and digital gaming respectively and have already been established in modern pedagogical and educational technology, consisting an undoubtedly promising field of research [256]. Nevertheless, there are different opinions about what the key game characteristics are. For example, Thornton and Cleveland [257] claim that *interactivity* is the most essential aspect of a game. Another study by de Felix and Johnson [258] suggested that the dynamic visual environment together with *rules* and *goals* of games are the essential features. Baranauskas et al. [259] stated that the heart of playing is *challenge* and *risk*. All these characteristics are summarized by Malone [260] in four key elements that define digital games:

- ✓ *Fantasy*, which stands for the scenario and the virtual world in which the activity is embedded.
- ✓ *Curiosity*, which is sustained by the continual introduction of new information and non-deterministic outcomes.
- ✓ *Challenge*, which is provided within each appropriate level of difficulty.
- ✓ *Control*, occurs during the game development and through the opportunities that players have to make choices which have direct consequences.

Prensky [256] argues that the prevalence of video games has actually rewired the way people think and has made traditional learning methods less effective. The author examines cognitive and pedagogical questions surrounding games and provides numerous case studies to clarify his points. To progress in a game is to learn since when a student is engaged with it, his mind is experiencing the pleasure of grappling with (and coming to understand) a new system [261]. This is true whether the game is considered entertainming, serious or educational. On one hand, serious games are those designed for a primary purpose other than pure entertainment and the idea explicitly emphasizes the added pedagogical value of fun and competition [262]. On the other hand, educational games are games explicitly designed with educational purposes, or which have incidental or secondary educational value. An educational game is a game designed to teach about a specific subject and to teach them a skill.

From the bibliographical research it is widely admitted that digital gaming can support learning in a rather positive way [263-266]. A wide number of significant research studies has shown the usefulness of GBL in education [267, 268]. The great advantage in using digital games in education is that they add a pleasant experience during the learning process and give extra motivation for learning [254, 255, 269]. Apart from being pleasant, they can be very effective and useful as plenty of research studies have already shown [267, 268, 270, 271]. Results show that the use of games in the curriculum help students experience how GBL can contribute to teaching and learning [272]. Especially in CSE many studies have used digital games as learning environments to support introductory programming. Hartness [273] developed Robocode, an open source game in order to support java programming. The game objective is to develop an artificial intelligence (AI) for a tank to fight against other tanks programmed by other players. Students simply develop their war strategy using programming and the battles run interactively when all players complete programming their own AI. CodeSpells [274] is another recent example of game, which aims to teach programming by the player being a wizard that can affect the environment through 'spells' coded in Java. Other examples of games that are specifically developed to teach programming is Catacombs [275], Saving Serra [275], Elemental [276] and more recently Prog&Play [277].

However, in practical terms, the design and deployment of digital games is costly in terms of time, effort, and money, mainly due to the graphical interface that they require and the narrative that is needed to support them. In this sense, gamification introduces a new approach which uses elements and dynamics of games with no ambition to deploy complex narratives or visual settings. Gamification is the idea of adding game elements (such as points, badges, leaderboards, competition, achievements) into a non-game setting [278]. While similar, gamification is a different breed of learning experience than GBL. Table 5 presents a comparison of the two methods. The approach has been successfully applied by commercial brands as a means of supporting user engagement and fostering user activity, social interaction, or quality and productivity of actions [279, 280]. Gamification has the potential to turn routine, boring tasks into refreshing, motivating experiences [278]. It can also use digital game design techniques and to embed elements like levelling curves and challenges into existing learning processes and apply thoughtful game design to promote real world objectives such as student engagement and better learning [281, 282].

In classroom settings, there have been several attempts to gamify learning activities with two main purposes: the first is to encourage desired learning behaviors, such as following software engineering best practices, promoting the participation of students into learning communities, or fostering active participation [281, 283, 284]; the second one is to engage students in learning, for example by the use of learning materials such as tutorials or digital tools [285, 286].

Table 5. Comparison of Gamification and GBL.

Gamification	Game-Based Learning
adds game inspired elements to meet the learning outcomes	use of games to meet the learning outcomes
applies game mechanics to a non-game environment	learning comes from playing games
incorporates badges, awards, achievements	use of commercial, serious or educational games
uses experience points (XP) as substitute for traditional grading	promotes critical thinking and problem-solving
provides students with choice in the learning path	allows students to experience the learning
can be accomplished by enhancing game elements in teaching	can be accomplished with the use of digital or non-digital games

Most of the educational gamified frames, use points and badges to reward progress and levels of expertise acquired by learners whereas the most extensive game dynamic used is competitiveness through leaderboards, for instance the Khan Academy[19] [284, 286]. Furthermore, gamification can be integrated in a more authentic manner as some classrooms may transformed into an interactive game. Such an example is Classcraft[20], a role-playing game supported by a digital platform and a mobile application that were developed to answer high school teachers' classroom management needs. Students can create their character, play as part of a team, and earn experience points and rewards based on class-related behaviors (Figure 10). Then, they get rewards for helping other students, producing exemplary work and likewise they can receive consequences for behaviors that are inconsistent with the desired the learning environment. Since it launched (August 2014), Classcraft has gained rapid usage by many teachers in many countries and languages which represents over 150,000 students connecting regularly to the platform [287].

Figure 10. Screen capture of the Classcraft game.

As educators, governments, and parents realize the psychological need and benefits that games have on learning, the GBL method has become mainstream. Games satisfy the fundamental need to learn by providing enjoyment, passionate involvement, structure, motivation, creativity, social interaction and emotion in the game itself while the learning takes place.

[19] Khan Academy - personalized learning with gamification characteristics: https://www.khanacademy.org/, last accessed December 2016
[20] Gamification through a digital classroom management tool: https://www.classcraft.com/, last accessed December 2016

3.4.9 Educational Robotics

Popular interest in robotics has increased surprisingly in the last few years since it seems that it offers many benefits in education, at all levels [288]. Yet, the commercial robotics market indicates that the growth for personal robots, including those used for entertainment and educational purposes, has been tremendous [289]. Educational Robotics (ER) is a teaching method that relies on the use of robots for teaching purposes [290]. ER is considered to be an approach to educational technology that offers many benefits in education at all levels [291, 292]. Through the construction of robots, learners can understand key concepts and ideas from all STEM related disciplines. As Rockland, et al. [293] underline, there is need to bridge the gap between the curriculum and the scientific nature of STEM-related concepts, by engaging students with alternative attractive learning approaches.

The primary objective of ER is to provide a set of experiences to facilitate learners' development of knowledge, skills and attitudes for the design, analysis, application and operation of robots [294]. The objective of using robotics in CS is to use them as a tangible and exciting application to motivate and facilitate the instruction of computer programming [295, 296]. Equally important, the introduction of robots in the classroom, both at school and university and even as extracurricular activity, promotes a host of values such as creativity, participation, support and teamwork [297]. ER can offer a unique alternative to traditional teaching methods by integrating STEM-based applications in CSE. Robotics are appropriate for introducing CS to young people because the students have a direct output of their programs and not just write code or see virtual characters on a screen [298]. Robots enable students to engage with CS concepts and programming in a fun and creative and they increase interest levels [299], motivation, problem-solving and student collaboration [300].

ER are known to support constructivist learning and provide the opportunities for the personal construction of knowledge and meaning [301]. ER's theoretical and pedagogical background is based on the field of Constructivism and the works of Piaget and Papert, where learning is viewed as a process that learners construct their personal meaning through active participation [98, 302, 303]. Papert [98] believed that robotics activities have tremendous potential to improve classroom teaching. He implied that learners do not learn from technology, but they learn with technology. Towards that direction, educators generate ideas and develop activities to incorporate robotics into the teaching of various subjects, including CS. Arlegui et al. [304] used a constructivist Problem-BL approach in order to teach key competences and standard curricula with robotics, at primary school level. Pittí et al. [301] show the importance of focusing research on pedagogical actions with ER features, which improve learning results. Another experience with ER [305] shows that an action-research approach, as a quite useful methodological tool can provide a systematic means to identify a problem based on the daily tasks of teaching.

Especially the educational robots LEGO® Mindstorms® (Figure 11) are directly founded in the work of Papert and are widely used for educational settings to teach programming, engineering [306]. They are also used to teach several skills, for example, LEGO® were used to teach students to be independent and achieve self-directed learning [307]. LEGO® Mindstorms® have been successfully used with students of different ages to support CSE from young children [308] to college students [309]. Interestingly, even when the same activities are used for primary school students and final year CS students there are significant learning benefits for both [310].

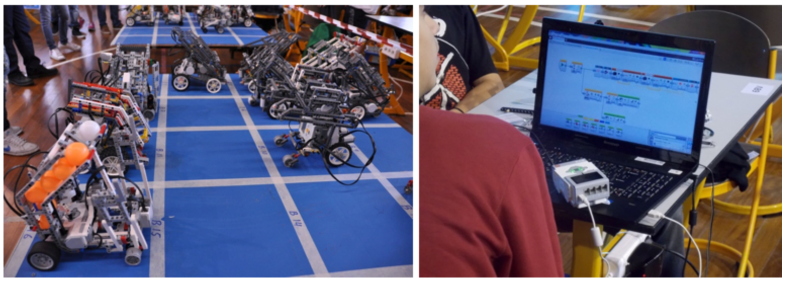


Figure 11. Lego Mindstorms Robots and their VPL.

In addition, Benitti [311] reviewed recently published literature on the use of ER, with a view to identifying the potential contribution of robotics as an educational tool, in the context of school education. In their study they summarized empirical findings and concluded that ER have an enormous potential as a learning tool. It seems that ER is a relevant tool for improving programming learning [312, 313]. Furthermore, ER enhances several learners' skills such as critical thinking [314], problem solving [313, 315] and communication [313]. However, the most important factor is that students use their powerful ideas, their own way of understanding, they represent their knowledge [292].

Next, these benefits many manufacturers develop robotic platforms for educational purposes. The following Table 6, was created in order to present the most recognized, per educational level that can be used and several other main characteristics:

Table 6. Educational robotic platforms.

Platform	Level	Features	Assembly	Programming	Open
Lego Mindstorms	Primary, Secondary, Vocational, University.	Its main component is a brick-shaped computer called the NXT Intelligent Brick AKA. It also uses several mechanical parts such as motors and sensors.	Uses ready-made blocks - LEGOs.	Has its own VPL (EV3), but also corporates with many programming languages like C, Python, Ruby and even the block-based Scratch.	no
ArcBotics Sparki	Primary, Secondary, University	Custom controller based on Arduino Leonardo Atmega32U4RC. It also uses several sensors and actuators.	Pre-assembled.	Arduino, C/C++, Ardublock, Minibloq, Chromebook (Codebender) and Python.	yes
Parallax BoeBot	Primary, Secondary.	Main controller, cube servos, sensor modules, joints, connectors, wheels, terminals.	easy slide & lock modules, expandable with LEGOs.	No skills needed for beginners, C for advanced users.	yes
EDBOT	Primary, Secondary, Tertiary.	Humanoid robot	Based on the Robotis Mini platform with Edbot software.	Scratch, Python, JavaScript and many others.	no
Fischertechnik	Primary, Secondary, Vocational.	Uses the Module Robot TX processor, with many other connections.	Construction ready-made toys.	Uses its own app "ROBO Pro", C compiler and Scratch.	no
Lego WeDo	Primary.	Uses a microprocessor brick component. motors and sensors.	Uses ready-made blocks - LEGOs	Drag-n-drop VPL.	no
ProtoCREA	Secondary, University.	Controller based on Arduino UNO with BT. It consists of a wheeled platform to learn design.	It has a 3D printed body.	Scratch4Arduino, Arduino, Ardublockly, BitBloq, Android & App Inventor, C++, ROS.	yes

Phiro	Primary, Secondary.	Controller - Atmel ATMEGA256. The SWISH technology helps in comprehending complex logical and loop statements.	Consists of a wheeled platform.	Scratch and Snap (for PCs) and Pocket Code (for Android).	no
Thymio	Primary, Secondary, Vocational, University.	Uses a micro-processor with sensors and accelerometer.	compact body, connection for other systems.	ASEBA scripting language.	yes
ROBOTIS DARwIn-OP	University	Anthropomorphic robots.	Pre-assembled.	Linux OS and many programming languages.	no

Concluding, ER has been introduced as a powerful, flexible teaching tool that stimulates learners to control the behavior of tangible models using specific programming languages (graphical or textual) and involves them actively in authentic problem-solving activities.

3.4.10 Non-textual programming

There is a long history of research on developing environments, tools and programming languages to help novices learn to program [316, 317]. Some of them have gained significant interest in the CSE community. These include:

- ✓ Narrative tools which support programming by creating and telling a story, like Alice [318] and Jeroo [319].
- ✓ Visual programming tools which support the construction of programs through a drag-and-drop interface, like Scratch [320], JPie [321], Alice [318] and Karel Universe [322].
- ✓ Flow-model tools which construct programs through connecting elements to represent order of computation, like Raptor [323], Iconic Programmer [324] and VisualLogic [325].
- ✓ Specialized realization that provides execution feedback in non-textual ways, like multimedia e.g. JES[21] [326], or kinesthetic robotics e.g. Lego Mindstorms [327].
- ✓ Tiered language tools in which learners can use more complex versions of a language as their expertise develops, like ProfessorJ [328] and RoboLab [329].

Particularly, the programming languages that have been developed to mitigate the various difficulties of programming education are known as Educational Programming Languages (EPLs) [330]. The task of specializing programming environments for novices begins with the perception that programming is a hard to learn [331]. Learners raise problems with many programming factors like looping constructs [332], conditionals [333], assembling programs out of base components [334] and there are probably other factors, and interactions between these factors, too. Therefore, EPLs employ a variety of techniques to ease the process of learning to program. These include:

- simplifying the programming language syntax [98, 335],
- matching more closely to the way that children describe the behavior of programs [336],
- enabling users to construct programs by assembling graphical tiles [320, 337-339] or filling in forms [340, 341],
- specifying programs as graphical rules [342], and

[21] Jython Environment for Students, is a free implementation of the Python programming language, http://python.org , last accessed March 2017.

- creating programs by demonstrating actions in a 3D animated world [343, 344].

All the above, allow students to focus on solving the problem instead of finding missing semicolons. Of course, the end goal is student learning and performance. Experiments on introductory programming environments have shown increased student learning [331]. Nevertheless, students' perceptions on EPLs show increased motivation and their positive attitude to learning computing [320, 345].

Moreover, with Visual Programming Languages (VPL) like Scratch, students can not only learn CS concepts, but also develop higher order thinking skills. As Fessakis et al. [346] state, computer programming can be seen as an important competence for the development of higher-order thinking, such as algorithmic problem solving skills. Research [347] revealed that students who perceived a flow programming experience, learn problem solving skills by applying trial-and-error, learning by example, and analytical reasoning strategies. Another point is that visual programming can make it easier to see the big picture [348]. The program is defined by its shape, so beginners and advanced users alike can get an idea of what the program does at a glance (Figure 12). Shapes and colors also take advantage of the user's visual abilities to a greater extent than code indentation and coloring ever could, to make the source more readable. This helps describe the program more easily. This also can help maintainability, since students can more easily see what the program it does, and how complex elements come together to make up it. A block structure also means that it is often easy to rearrange the graphical code, as opposed to the tedious refactoring of text-based source code.

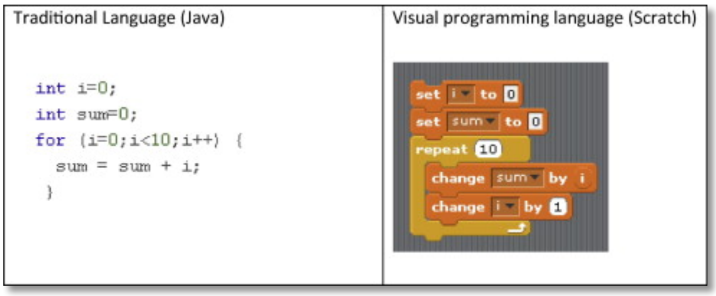


Figure 12. Coding sample on a text-based language and on a VPL.

The following Table 7 was created in order to provide a comprehensive list of the most recognized VPLs that are used for educational purposes including a categorization about their main characteristic and educational level that can be used.

Table 7. Visual Programming Environments for Education.

VPL	Categorization	Description	Ages - Grades
AgentSheets	block based / drag-and-drop	Scalable Game Design: programming through game design and authoring	middle and high school
ArduBlock	block based / drag-and-drop	graphical programming for Arduino	middle and high school
Alice	block based / drag-and-drop	object –based educational programming with an IDE which uses/creates 3D storytelling	for all ages
App Inventor	block based / drag-and-drop	tool for creating apps for Android based on Blockly	for all ages
Blockly	block based / drag-and-drop	project by Google that uses visual block programming editors and runs in a web browser	apps that use Blockly are suitable for all ages e.g. Code.org
Blueprint	data flow	a complete gameplay scripting system based on the concept of using a node-based interface to create gameplay elements from within Unreal Editor	higher education

Etoys	block based / drag-and-drop	media-rich authoring environment with a scripted object model for many different objects	young children - primary education
Flowhub	flowcharts	building full-stack applications in a visual way with a flow-based programming environment	for all ages
Lava	block based / drag-and-drop	visual object-oriented, interpreter-based programming language with an associated programming environment that uses structure editors	for all ages
Mama	block based / drag-and-drop	object-oriented programming designed to help young students start programming by providing all language elements in the student mother tongue	primary and secondary education
TouchDevelop	block based / drag-and-drop	a Microsoft Research interactive development environment for mobile devices	for all ages
Minibloq	block based / drag-and-drop	graphical environment for Arduino that is used especially in robotics	middle and high school
Flow Graph	data flow	visual scripting system that is embedded in the CryENGINE Sandbox Editor and users do not need to have any scripting or programming knowledge	high school and above
Flowgorithm	flowcharts	graphical authoring tool that creates executable flowcharts which can be converted to several languages	high school
Grasshopper 3D	data flow	programs are created by dragging components onto a canvas while the outputs to these components are then connected to the inputs of subsequent components.	high school and above
Hopscotch	block based / drag-and-drop	create games, art, and more in a creative way for mobile touchscreen devices (used on iPad)	for all ages
Kodu	event-based	3D visual programming tool from Microsoft FUSE labs, which builds on ideas begun with Logo	for all ages
LabVIEW	data flow	a system-design platform and development environment from National Instruments which uses a graphical language named "G" and commonly used for data acquisition, instrument control, and industrial automation on a variety of operating systems	higher education
Node-RED	flowcharts	developed by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things	high school and above
Open Roberta	block based / drag-and-drop	cloud-based programming environment designed for use with Lego Mindstorms robotic kits	for all ages
Quartz Composer	data flow	a node-based VPL provided as part of the Xcode development environment in macOS for processing and rendering graphical data	higher education
Raptor	flowcharts	programs are created visually and executed visually by tracing the execution through the flowchart	middle school
Scratch	block based / drag-and-drop	a product of MIT designed for kids in K-12 and after school programs	from 8 to 16
Simulink	data flow	a graphical programming environment for modeling, simulating and analyzing multidomain dynamic systems	higher education
Snap	block based / drag-and-drop	an extension of Scratch with first class procedures and lists which is used for teaching by UC Berkeley (browser-based)	higher education
Stagecast Creator	block based / drag-and-drop	a tool based on the programming by demonstration concept, where rules are created by giving examples of what actions should take place in a given situation	primary and secondary education
StarLogo	block based / drag-and-drop	an agent-based simulation language developed at MIT Media Lab. It is an extension of the Logo programming language, a dialect of Lisp	primary and secondary education
ToonTalk	block based / drag-and-drop	system intended to be programmed by children in the form of animated characters, including robots that can be trained by example	primary and secondary education
Visual Logic	flowcharts	tool that creates executable flowcharts	for all ages

Moreover recently, tangible user interfaces (TUIs) have been developed to provide more intuitive interactions in computing environments [349, 350]. TUI programming tools help young novice programmers complete programming activities without the worry of learning tool usages by using intuitive handling, such as connecting and stacking command objects [351]. TUI can be beneficial for learning syntax and commands easier than a traditional language and are very effective for young children [352, 353]. Furthermore, the successful transitioning between theory and practice could be effective when using a tangible programming language [351]. Finally, more interactive methods, like tangible programming, are effective when the goal is higher-order thinking and developing a deeper understanding [139, 354].

It is clear that there are many alternatives when an educator comes to deal with several difficulties and limitations of teaching programming. Knowing what exist and what are its main advantages could be very helpful in making decisions and helping students.

3.4.11 Contextualized Learning

A contextualized computing course is one in which one or more application domains provide the motivation for learning CS content and inspire the design of learning activities; these domains may be, and often are, external to CS itself [355, 356]. Contextualized Learning always take place in a context, whether we build on that context or ignore it. We can either recognize this, use context as a tool, and make it highly effective in supporting student learning or ignore it and hope for the best. According to Cooper and Cunningham [356] the adoption of a context for teaching CS eases the way for students to work with, and understand, the traditional content of CS concepts. The fundamentals are simply introduced and developed through the use of the outside examples and motivation and therefore the concept of context gives the opportunity to build on the strength of the program or of individual instructors in selecting a context and deciding how to use it.

Following in the constructivist tradition, contextualized learning may increase learning and motivation, and eventually increase the number of students who are attracted to CS and seek to remain in the field [355, 356]. The method has been tested in several learning contexts [357, 358] and it gives students a good background for building programming skills which seems to be very successful in attracting and retaining them in CS. Researchers used computer graphics in introductory programming courses [359, 360] and noticed that the this context captured students' imagination and made them more creative. There is also evidence that contextualization can help improve gender balance in computing [361].

Nevertheless, contextualizing a course may require compromises in the CS content covered, since learning about a context takes additional time. Moreover, contextual approaches run the risk of inhibiting transfer due to an excessive focus on the chosen context [362]. It can be difficult to choose a context that suits a varied group of students. Since there is a wide applicability of computing and wide variety of learners interests and communities of practice, there are many approaches to contextualizing computing content, ranging from explicitly contextualized curricula to individual teachers' implicit, ad-hoc contextualization of individual learning activities [355].

A well-known contextualized approach is Media Computation (MC) developed at Georgia Institute of Technology by Guzdial [363]. The critical characteristic of MC is that students create expressive media by manipulating computational materials (like arrays and linked lists) at a lower-level of abstraction. They can manipulate images by changing pixels, create sounds by iterating over samples, render linked lists into music, and create artifacts like collages, music, and digital video special effects. By that way, students learn CS concepts. Through MC learners have significantly lowered dropout rates and the gender gap is reduced in introductory courses for non-majors [40, 326, 364]. In addition, there is evidence that the method provides highest understanding even among different backgrounds of learners [40, 365].

Lukkarinen and Sorva [366] in their review about contextualized programming education and MC especially presented a classification of the tools used for each case (Figure 13 and Figure 14). The "Tools" category encompasses a broad spectrum of technological tools that students may use as a part of contextualized programming and the authors identified a number of subcategories as shown in Figure 13.

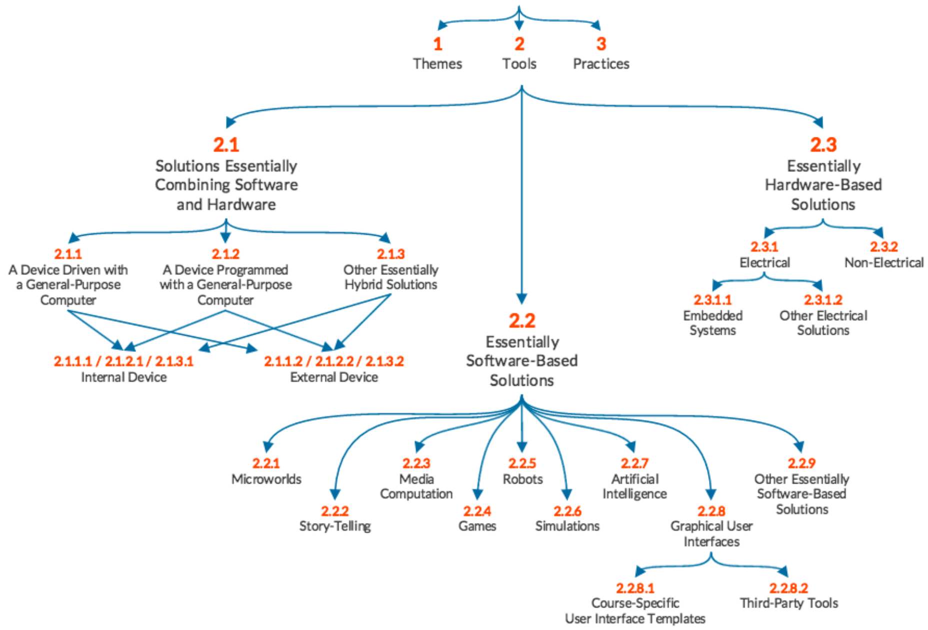


Figure 13. Tools for Contextualized Programming Education [366].

Figure 14 illustrates Lukkarinen and Sorva [366] classification of tools for MC. These tools are first divided into Program Code Libraries (branch 1) and then into Integrated Development Environments (IDE) (branch 2), of which the latter includes not only fully-fledged IDEs but also text editors, ides, worksheets, Read-Eval-Print-Loops (REPLs), and comparable tools.

Their report illustrates that there are numerous paths available to the programming for teacher interested in contextualization and media computation.

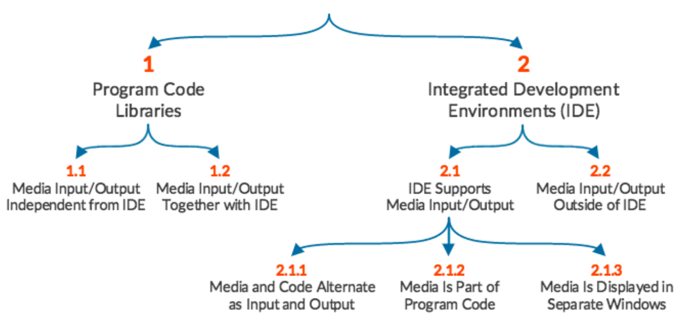


Figure 14. Tools for Media Computation [366].

These results can help educators and educational technologists who are interested in learning about MD and contextualized programming education, building new tools, and positioning their work in relation to the literature.

3.4.12 CS Unplugged

The CS Unplugged project provides ways to expose students to ideas from CS without having to use computers [367]. This has a number of applications, including outreach, school curriculum support, and clubs. The project provides a collection of resources for CS outreach and teaching on its web site, csunplugged.org[22], all are available online at no charge. This unusual approach introduces students to CT through concepts such as binary numbers, algorithms and data compression, separated from the distractions and technical details of having to use computers. In fact the unplugged approach presents fundamental concepts in several CS areas such as networks, artificial intelligence, graphics, information theory, Human Computer Interfaces, and of course programming languages. Bell et al. [367] suggest that no programming is required to engage learners with CS concepts and they propose a series of games and puzzles that use cards, string, crayons and lots of running around in order to accomplish that.

CS Unplugged is suitable for people of all ages, from elementary school to seniors, and from any background. The activities book[23] is translated in 17 languages including Greek. Therefore, in the last few years it has a large international uptake. Initially, it gained visibility after being added to the ACM recommendations for the K-12 curriculum [11]. Thereafter, the audience for CS unplugged has grown over the years and although it was intended for outreach, in some countries it is started to be used as teaching material in the curriculum.

Since the activities do not depend on computers, this avoids confusing CS with programming or learning application software. It also makes the activities available to those who aren't able to or don't want to work with computers, and skips the barrier of learning to program before being able to explore ideas. The unplugged approach even helped a jury of lay people understand CS concepts during a patent infringement case.

The activities tend to be kinesthetic, often on a large scale and involving team work. For example, the Sorting Network activity (Figure 15) has teams of six running through a network drawn on the ground and aims to demonstrate parallel sorting of networks.

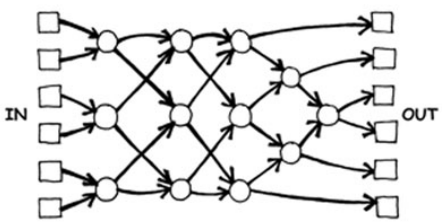


Figure 15. Sorting Networks - CS Unplugged activity.

The activities tend to allow students to discover answers for themselves, rather than just being given solutions or algorithms to follow. A constructivist approach is encouraged, as students to realize that they are capable of finding solutions to problems on their own, rather than being given a solution to apply to the problem. For example, students don't really need to be able to convert numbers to

[22] CS Unplugged Project: http://csunplugged.org/, last accessed January 2017
[23] CS Unplugged activities book: http://csunplugged.org/books/, last accessed January 2017

binary, but it is valuable for them to discover the patterns such as the doubling value of bits, patterns when you count in binary, and how the range increases exponentially as you add bits (Figure 16).

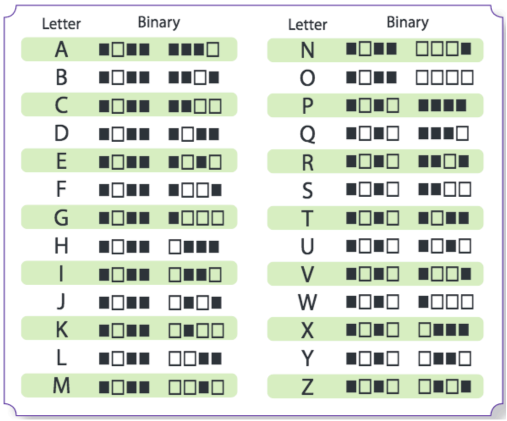


Figure 16. Depiction of ASCII Encoder Card for practice encoding simple data, like letters.

The approach of learning computing without computers is fun and engaging. Unplugged activities leave students with a sense of genuine achievement. There is often a strong sense of story in the activities; problems are presented as part of a story rather than as an abstract mathematical challenge. Children are more interested in pirates than privacy, and absurd fictitious stories can be more memorable than compelling business applications. In addition, the activities use equipment commonly found in classrooms easily at no cost. Most require only paper and pencil, and perhaps cards, string, chalk, whiteboard markers, balls or similar items.

Taub et al. [368] in their study examined the effect of the activities on middle-school students' views of CS Unplugged activities. Their results indicate that 'although the students generally understood what CS is' they perceived the computer as the essence of CS and not primarily as a tool, contrary to the intention of the CS Unplugged activities. They reported that it is difficult to demonstrate that the unplugged approach actually achieves its wider long-term goals.

Such a teaching method promotes CS to students as an interesting, engaging, and intellectually stimulating discipline. The activities capture learners' imagination and address common misconceptions about what it means to be a computer scientist [369].

Finally, such activities have the big advantage that will still be fresh after many years since they are fundamentals that do not depend on particular software or systems.

3.4.13 Subgoal Learning

Subgoal learning has been used to help learners recognize the fundamental structure of the procedure being exemplified in worked examples [370-372]. Subgoal-oriented instructions are typically implemented as worked examples. Worked examples give learners concrete examples of the procedure being used to solve a problem [372]. To promote deeper processing of worked examples and, thus, improve retention and transfer, worked examples have been manipulated to promote subgoal learning. Subgoal labeling is a technique used to promote subgoal learning and is used in the fields of cognitive science and educational psychology. The technique is about giving a name to a group of steps, in a step-by-step description of a process, to explain how the group of steps achieve a related subgoal (Figure 17). Lower-level steps of a worked example are grouped into a meaningful unit and labeled. Therefore, subgoal labels are function-based instructional explanations that describe the purpose of a subgoal, or functional piece of the problem solution, to the learner. They have been used in worked examples to teach learners to solve problems in STEM domains [372].

Learning with subgoal labels improves programming learning [373]. It also produces higher learning gains and better problem solving performance than learning without subgoal labels [374]. The underlying structure of the labeling (e.g. worked examples) promotes self-explanation [375, 376], and greater number of self-explanations are related to more successful learning. Subgoal labeling helps learners identify the structural information from incidental information and can reduce cognitive load because the learner has fewer possible problem-solving steps to focus [377]. The subgoal-labeled worked examples might provide learners with mental model frameworks. In a recent study [378], novices were given labels for subgoals and they mentally organized the information by using those labels when explaining how they solved a problem. Authors suggest that subgoals aid learners in developing mental models early in the learning process which reduces the extraneous cognitive load imposed on novices learning programming. Subgoal labels reduce cognitive load and is a type of guided instruction that could give learners a framework for a rational mental model that they could have filled in with information [379].

No labels	Given Labels

sum = 0
lcv = 1
 	
WHILE lcv <= 100 DO
 	
sum = sum + lcv
 	
lcv = lcv + 1
 	
ENDWHILE
	<u>Initialize Variables</u>
	sum = 0
	lcv = 1
	
	<u>Determine Loop Condition</u>
	WHILE lcv <= 100 DO
	
	<u>Update Loop Var</u>
	lcv = lcv + 1
	ENDWHILE

Figure 17. Example of labeling into a text based program.

Margulieux et al. [378] inserted subgoal labels into App Inventor videos (Figure 18). Their study suggests that labels led to improved learning, retention, and even transfer to new App building problems, all compared to the exact same videos without the subgoal labels.

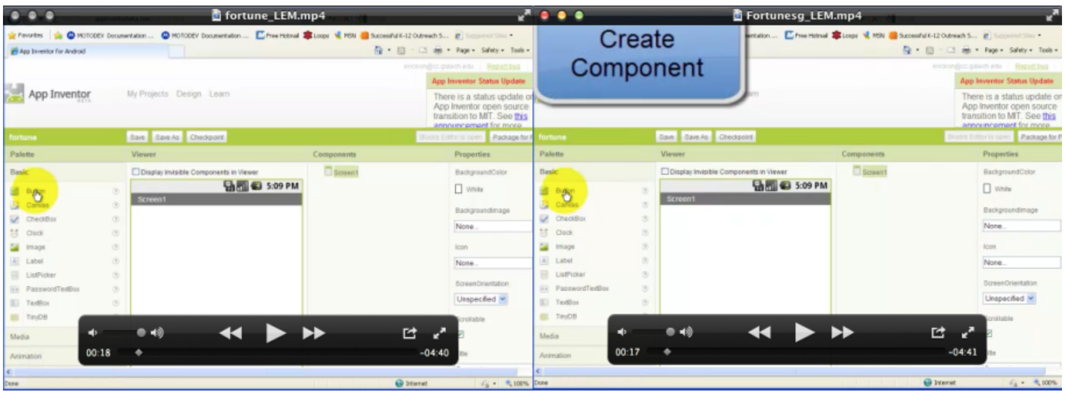


Figure 18. Subgoals in App Inventor by video form.

The subgoal intervention manipulates the instructional materials that students receive and therefore, reaching a large number of students with the work of a small group of people would be relatively easy. [380]. These interventions are not dependent on instructors and therefore they can also be used in learning environments without personal interaction with an instructor, such as online learning. Instructional text with subgoal labeled worked examples can further improve learners' performance in problem solving in a computer-based learning environment [378]. Subgoal learning can be used in different areas such as teaching and learning novel problem solving, in training teachers to teach technical subjects, multi agent programming, professional development, online learning and other types of lifelong learning.

3.4.14 Parson's Programming Puzzles

Parson's Programming Puzzles [381] (or else Parson's or Parsons Problems) is a method that provides practice with basic programming principles in an entertaining puzzle-like format. Careful design of the items in the puzzles allows the instructor to highlight particular topics and common programming errors. Since each puzzle solution is a complete sample of well-written code, use of the tool exposes students to good programming practice.

Parson's puzzles are simplified code construction assignments where the lines of code are given in the wrong order and the task is to sort and possibly select the correct lines. Originally, Parson's puzzles were developed to provide an engaging learning environment with immediate feedback. They are widely used and there are tools supporting them.

Automatic assessment of Parson's puzzles is straightforward as it can be done without executing the code. In addition to online learning environments, Parson's puzzles can be used in traditional paper exams where coding exercises (i.e., programming with a pen and a paper) are often problematic. Interestingly, in the context of paper exams, points from Parson's puzzles correlate well with open ended code writing question [382] (Figure 19).

```
return result;
return word;

String result = "";
String result;

if (word.charAt(i) == 'a')
if (word.charAt(i) != 'a')

for (int i = 0; i < word.length(); i++)
for (int i = 0; i < word.length; i++)

result = result + word.charAt(i);
result = word.charAt(i);

private String removeAllAs(String word)
private String removeAllAs(word)
```

Figure 19. Example of a paper-based Parson's Programming Problem.

Moreover, in the same study, points from neither of these question types correlated with points from tracing exercises. However, in another setup, Lopez et al. [143] found Parson's puzzles to be lower level than tracing exercises including loop constructs. Lopez et al. speculate this could be due to different difficulty levels or complexities of tracing and Parson's puzzle exercises in their study.

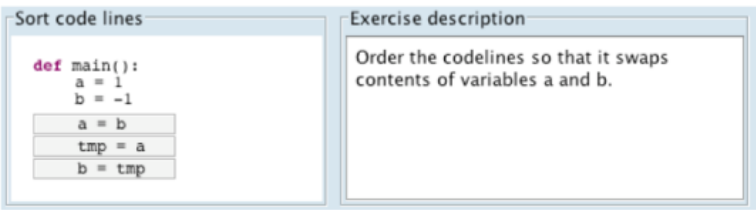


Figure 20. Example of an automated Parson's Programming Problem.

Parson's problems, provide all the code required to solve a given problem, but require the students to order (and possibly select then order) the lines of code to form a correct solution.

Research on Parson's problems suggests that they might be a more effective and efficient learning approach than writing equivalent code, especially for time-strapped secondary teachers [383].

Parsons problems appear to effectively assess similar skills that traditional code-writing questions assess, and at the same time are considerably easier and less subjective to mark [382]. On weak students the Parsons problem format provides an opportunity to guess an answer that is not available in code-writing questions. Furthermore, Parson's puzzles help students with syntactic problems of programming languages and recognize patterns more easily. According to researchers [384, 385], many novice programmers have problems in reading and in writing programs. One explanation is that novice programmers miss programming schemas or plans [386]. These kind of programming puzzles help learners to recognize those patterns immediately and to apply and combine them when writing programs.

3.4.15 Extreme Programming

In recent years, the growth of extreme programming (XP) has brought considerable attention to collaborative programming. In software development, the term 'agile' is adapted to mean the ability to respond to changes, either they have to deal with Requirements, Technology or People [387]. XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software [232]. It is one of the Agile software development methodologies and was developed by Smalltalk code developer and consultant Kent Beck and his colleagues, Ron Jeffries and Ward Cunningham [232]. Moreover, XP is an approach that credits much of its success to the use of Pair Programming (PP), regardless of the users experience [236]. There are four basic activities in Extreme Programming [232]:

- ✓ Coding
- ✓ Testing
- ✓ Listening
- ✓ Designing

These four basic activities comprise of several practices to achieve that the XP objective (Figure 21). Wherever one of the practices is weak, the strengths of the other practices will make up for it.

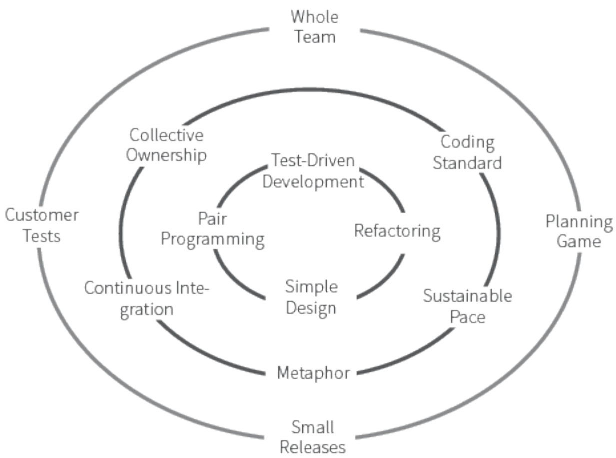


Figure 21. Extreme Programming - Practices [232].

The evidence of XP success [232] and other light or agile methodologies [387] is so impressive that it has aroused the curiosity of many researchers and consultants from education. Andersson and Bendix [388] developed a method of teaching that they call eXtreme Teaching (XT), inspired by XP. The XT method highlights the importance of values such as feedback, communication, respect and courage and is highly iterative while its nature resembles Kolb's learning cycle (Figure 6). Nevertheless, XT approach demands a lot from the instructors, who are the main instruments of putting it to action. Despite of being demanding in a sense, it can increase dialogue between the involved teachers and the students. Students feel more committed to the course as they become more involved in it [388].

XT combines fundamental concepts of teaching with practices from XP, for example, the idea that teaching in itself is the product in this context. Instructors, aim at the best possible product, such as student passing their exams and getting a degree [388]. The idea is that nowadays learners, are both customers and users of teaching [232]. They know their own rights, and capabilities or limitations, and the value of this should be put in the use in getting them involved. In higher education traditions, students are usually approached as teaching subjects [388] while in the XT approach, not only the teachers but also the students can be developers of the teaching product. The instructor's role involves not only teaching but also coaching. Astrachan et al. [389] adapted principles of XP in classroom teaching. Although academic requirements, goals, and methods differ those in industry, the authors found that many aspects of XP can be incorporated into the design and implementation of a university-level CS and programming curriculum. Similar to XP, all programming in XT is built by two learners (PP), sitting side by side, at the same machine. This practice ensures that all production code is reviewed by at least one other, and results in better design, better testing, and better code. XT encourages students to take an active part into the learning process [388]. They also may become more motivated as they feel that they could really contribute to the course [390].

Software development is a complex area and extensive practice and reflection is required in order to obtain a good understanding of the different tasks involved in going from idea to deliverable software [391]. XP has been used as a vehicle for teaching the basic concepts in software engineering. Hedin et al. [391] developed a team programming course based on XP. The authors discuss how these courses affect the curriculum, and how they used the approach to support the learning cycle through the use of iterative learning, peer learning and a fixed-time budget. XT is a method that helps to give the students a solid basic understanding of not only the important concepts mentioned above, but also of, e.g., iterative development, configuration management, and team communication. In particular, the highly iterative nature of XP, where the participants get rapid feedback at all levels, makes it ideal in a teaching situation [392]. As a side effect, the PP also gives the students practical skills in testing, configuration management, and refactoring, skills that they can immediately use in their projects in later courses.

3.4.16 Program Visualization

Program Visualization or else Visual Program Simulation (VPS) is a pedagogical technique for introductory programming education. In VPS, a student takes on the role of the computer as executor of a given program. The student uses a visualization of an abstract computer, a notional machine, as an aid to illustrate what the computer does as it processes the program. The goal of the VPS activity is to help the student learn about programming in general and about specific programming concepts. Du Boulay [393] invented the term notional machine for "the general properties of the machine that one is learning to control" as one learns programming. He identifies the notional machine as one of the major sources of difficulty for novice programmers.

There are many software systems for program visualization in introductory programming education. Jeliot [394] is a program visualization system for novice programmers. It automatically generates visualizations of the step-by-step execution of Java programs within an abstract computer. The user can write any Java program and have Jeliot execute it visually. Jeliot uses animation to display steps such as assignment, expression evaluation, passing parameters, returning values, etc. Based on earlier studies, [395] argue that higher levels of student engagement with a visualization lead to more effective learning. UUhistle [396] is a highly interactive program visualization system for introductory programming courses with many VPS exercises, in which the student takes the role of the computer as executor of a program in order to learn about program dynamics and to demonstrate their understanding. The student needs to predict the steps involved in the execution of a given program and to directly manipulate the elements of a program visualization to show what happens next. The software according to the authors aids learning by promoting a high level of student engagement with visualizations. Moreover, DAVE [397], is an interactive dynamic algorithm visualization system for the introductory lessons in algorithm design and programming. DAVE allows students' experimentation not only with sample algorithms, constructed by the designer, but mainly with the automatic animation of their own algorithms. This software, may offer to introductory courses about algorithms and programming not only by facilitating constructivist and discovery learning activities but also by supporting different types of learners.

Furthermore, VPS is a program-reading activity in which the student traces the execution of a given program rather than creating a program of their own [398]. Reading code is increasingly being recognized in the literature as an important component of introductory programming education, to be learned before or in parallel to program writing [399]. A program-reading task such as a VPS exercise can be used to provide an interactive worked-out example of program writing; judicious use of examples helps manage learners' cognitive load during the learning process. VPS is understood as being related to how computers execute programs, and that in turn is understood to relate to how one reads and writes computer programs as a programmer [396].

Sorva [398] in his doctoral thesis investigated VPS and provided a theoretical framework for it and presented a software system that facilitates the use of VPS in practice. The author argued that VPS can help learners discern the crucial dynamic aspect to programs and programming, and to construct a better mental model of the mechanisms underlying program code. In this way, VPS can help novice programmers across a key early threshold. Practice using VPS serves to ingrain the mental model and make it a natural, efficient part of the learner's thinking.

3.4.17 Competency-Based Learning

Competency-Based Learning (CBL) is an approach to teaching and learning that gives importance in concrete skills than abstract learning. Competency-based strategies provide flexibility in the way that knowledge is earned, and provide students with personalized learning opportunities [400]. These strategies include among others situated learning, online and blended learning, flipped learning and learning by taking account personal characteristics. The CBL method has its roots in teacher education [401, 402]. However, there is no one specific thing that characterizes it and researchers often use many related terms in different ways that causes confusion about CBL and Competence-

Based Education (CBE). Such terms are adaptive learning, differentiated learning, individualized learning, do-it yourself education and self-paced learning.

This section looks at the CBL term by a more abstract point of view, where teaching and learning can use different techniques and tools for the learners' service. Therefore, the most important characteristic of CBL is that it measures learning rather than time. Students show their progress by demonstrating their competence, which means they prove that they have mastered the knowledge and skills (competencies) required for a particular course, regardless of how long it takes [403]. While more traditional models can and often do measure competency, they are time-based and students may advance only after they have put in the seat time. Broadly, the CBE is a form used in higher education settings in which credit is provided on the basis of student learning rather than in addition to the number of credit or time spent in class, because it has the potential to lower costs and serve adult learners in search of flexibility [404]. However the CBL method is particularly suited to CS due to the nature of the discipline where well-defined, measurable outcomes can be designed and implemented [405].

The appeal of CBL is obvious. It focuses on attaining and demonstrating specific elements of domain knowledge. Policy makers and researchers in education have recognized the benefits of competency-based approaches. This type of learning leads to better student engagement and motivation because the content is relevant to each student and tailored to their unique needs [406, 407]. It also leads to greater understanding and better student outcomes because the pace of learning is customized to each student [408]. Moreover, CBL leads to increased student retention and completion rates, particularly when prior learning can be applied to the learning progress [405]. Learners' also show improved ability to recognize, manage, and continuously build upon their own competencies and evidence of learning. Finally, in CBL courses, learning resources, and assessments are aligned to well-defined goals and therefore, instructors improve their ability to understand learners' competencies and learning achievements [400].

Moreover, learners have different objectives and predispositions. Hence, an optimal learning path for one learner is not necessarily the same for the others. That is the key element of Personalized Learning (PL) method which seeks to accelerate student learning by tailoring the instructional environment to address the individual needs, skills and interests of each student. Students can take ownership of their own learning, while also developing deep, personal connections with each other, their teachers and other adults [409]. To accomplish this goal, schools, teachers, guidance counselors, and other educational specialists may employ a wide variety of educational methods, from intentionally cultivating strong and trusting student-adult relationships to modifying assignments and instructional strategies in the classroom to entirely redesigning the ways in which students are grouped and taught in a school. Because PL has such broad implications, and the term encompasses such a wide variety of potential programs and strategies, it may be difficult to determine precisely what the term is referring to when it is used without qualification, specific examples, or additional explanation. However, the approaches described in this section rest on the notion of competency, which has been formalized for implementation with the use of e-Learning environments [410].

PL term, or personalization, refers to a progressively student-driven model consisted of a diverse variety of educational programs, learning experiences, instructional approaches, and support strategies that are intended to address the distinct learning needs, interests, aspirations, or cultural backgrounds of individual students [411].

Researchers have been engaged in the development of educational computer games by taking personal learning characteristics into account. Hwang et al. [412] proposed a personalized game-based learning approach based on the sequential/global dimension of the learning style [129]. To evaluate the effectiveness of the proposed approach, a role-playing game was implemented based on the approach. Authors found that the personalized educational computer game not only promotes learning motivation, but also improves the learning achievements of the students [412].

Although the term PL can be used in several contexts, it has become more widely in online education and learning programs. PL can differ significantly from the forms of the method being offered and promoted by virtual schools and online learning programs. In some sets, however, it can take the form of Blended Learning (BL) or the practice of using both online and in-person learning experiences when teaching students. BL is an instructional model that combines the teacher-led instruction with the use of technology and face-to-face instruction with computer-assisted instruction [413]. There are many strategies for using and delivering educational content online, such as having instructors prepare online lectures, wrapping the course around a massive open online course (MOOC), and curating online videos from various sources. Online education has spawned a rich literature on the effectiveness and efficiency of various forms of electronic teaching tools, from full online courses e.g., [414] to web-assisted, lecture-based courses e.g., [415, 416].

Table 8 presents the main differences between the traditional learning and BL methods [417]. BL can offer great advantages for both learners and instructors. The option of online lecturing is popular in terms of allowing students to work more slowly through the material and replay it and for those who need more reinforcement of concepts and for revision prior to exams [418]. Moreover, the traditional lecturing is enhanced by additional support that would not otherwise have been possible. Research provides evidence that there is an improvement in learning efficiency [419], although students with access to online resources do not necessarily achieve better learning outcomes [420]. Yigit et al. [421] used BL in programming courses in Computer Engineering Education. The authors show that education settings with BL can be more effective since students' achievements are better than in traditional education although CT abilities of students was found to be very close in both settings. Another study by Wang et al. [422] showed that BL provides great flexibilities to both teaching and learning of programming. Students that used BL achieved greatly improved results in computer programming courses and yet the face-to-face teaching process was easier for instructors.

Table 8. Differences between traditional and blended learning [417].

Feature	Traditional Learning	Blended Learning
Location	Physical classes (not flexible)	Anywhere (flexible)
Method	Face-to-face	Face-to-face & Online
Time	At specific time (not flexible)	Any time (flexible)
Technology	Not essentially	Use of technology

Finally, researchers and developers actively seek technological interventions that can greatly increase interactivity in BL. Shen et al. [423] developed a cutting-edge mobile learning system that can deliver live broadcasts of real-time classroom teaching to online students with mobile devices Their system allows students to customize their means of content-reception, based on when and where the students are tuning into the broadcast.

Another form of using the online content to the teaching and learning processes is Flipped Learning (FL). FL is a pedagogical approach in which direct instruction moves from the group learning space to the individual learning space, and the resulting group space is transformed into a dynamic, interactive learning environment where the educator guides learners as they apply concepts and engage creatively in the subject matter [424]. The basic principles (Figure 22) of the FL method are to deliver content outside of class settings and to move active learning into the classroom. In FL students come to the class after completing significant preparatory work, so that they can participate in the class activities. This prep work typically includes learning materials such as lectures and presentations delivered via podcasts or video-on-demand services [425, 426].

FL allows for a variety of learning modes. Instructors often physically rearrange their learning spaces to accommodate a lesson or unit, to support either group work or independent study. They create flexible spaces in which students choose when and where they learn. Furthermore, instructors who flip their classes are flexible in their expectations of student timelines for learning and in their assessments of student learning.

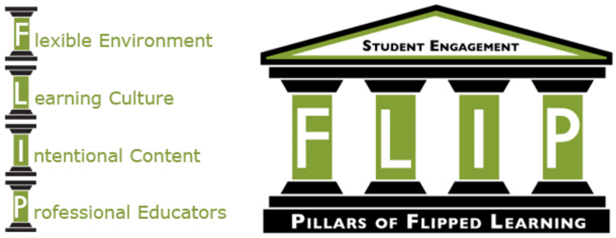


Figure 22. The four pillars of Flipped Learning, F-L-I-P ™ [424].

Concluding this section, CBL can be valuable for all of the stakeholders in learning communities: learners can have more opportunities to take ownership of their learning and expand their lifelong learning pathways; instructors can grow professionally as they articulate the learning outcomes in their areas of expertise and embed them in rich learning experiences; academic researchers can provide engaging curricula that advance knowledge and produce students who can demonstrate what they've learned; and curriculum designers focus on new ways of identifying barriers to success and achieving improved outcomes.

3.4.18 Social Networks for Assisted Learning

With the rapid development of the Web 2.0 technologies, Social Networks (SN, e.g. Facebook, Twitter, and Wikis) develop rapidly and become more and more popular [427]. The rise of SNs brings new possibilities in many aspects, including education. Students can share and exchange resources, ideas, and product easily, and create more information together with others via SNs. Using the Ellison [428] definition , a social network is a "*web-based service that allows individuals to (1) construct a public or semipublic profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the syste*m"(p.211).

Previous studies show an extensive use of social media by teenagers that seem to spend a large proportion of the day logged on. In the USA average use per person per day was about 6,5 hours [429]. In addition, college students spend more time on SN rather than in classes [430]. More than 95% of British undergraduate students use SN regularly [431]. Even in cases where the use of SN is not allowed, a common reaction of formal educational systems [432], students discover new ways to overcome the restrictions [433, 434] and therefore, since they will use them, formal education might need to consider incorporating them in teaching and learning, and transform them in learning tools. However, SN are rarely used in formal education [435].

From a constructionist point of view, the very characteristics of SN could directly lead to learning, since they support easy exchange of information, communication of learners, social connections, etc. [102, 436-439]. There have been past attempts to investigate the role of SN in formal education [440, 441]. For example, Berg et al. [442] used SN in a higher education setting to build better relationships between the university, students and staff. Another study found that teachers do not have directly negative attitudes towards the use of certain social software [443]. However, it is a known phenomenon that teachers and faculty usually react to the use of new technologies in class and they rarely adapt new tools effectively [444]. Especially, regarding SN and Facebook, they seem to believe that it is not for educational purposes [445]. On the other hand, students seem ready to experiment and use new technologies for learning purposes [444, 446], including SN and Facebook [445] and in fact, students spend a large proportion of their time in SN talking about their education and specific learning activities [447], although they do not view SN as a primarily academic tool [446]. Thus, research is still limited and there are many issues to be clarified. For example, student behavior is culturally affected [448-451] but most studies regarding SN in formal education have been conducted in English speaking areas [446] and to our knowledge no such attempt exists in Greece. Thus, both for issues of possible cultural differences but also due to knowledge gaps in this domain, the present study was carried out.

Furthermore, at least on a theoretical level, SN seem to have some desirable attributes that could be beneficial to learning [452]. A constructionist approach to learning emphasizes the importance of social aspects in the learning processes and SN seem to be able to support socialization and communication of users - potential learners [453, 454]. Again at a theoretical level, the use of technology and social software can vary from being viewed as means to replace traditional schooling [455] to simply being educational tools that add to the traditional learning. The authors of the present work, view social software as an educational tool with considerable potential, that does not simply add to the traditional learning processes but it can change the very nature of learning, since technology is not a neutral medium [456]. In addition, SN can be a teaching tool in formal educational settings, without being able to replace traditional schooling (at least yet).

Finally, there is a specific type of SNs that result from interaction between learners, teachers, and modules of learning [457] called Social Learning Networks (SLN). Such an example is the Edmodo[24] platform. Research has shown that the use of the Edmodo as a learning tool had a positive impact on the students and that it can be used as an educational tool to help engage students more in the use of more social media networking sites [458].

3.4.19 Emerging Technologies

New technology offers the opportunity for learners and instructors to communicate easier and to call on information and knowledge [459]. For many years, educators have looked towards emerging technologies to bring about a transformation in education where students are engaged in rich authentic experiences that will enhance their motivation and learning. Intuitive technologies, have been identified in the Horizon Report [460] as being the emerging technologies likely to have a significant impact on learning in K-12 schools.

[24] Edmodo SLN platform: https://www.edmodo.com/ , last accessed March 2017.

Tablet computing reveals a strong uptake in education since it has become a "*pervasive part of everyday life in much of the world*" [461]. Portability, the touch screen feature, ease of use, long battery life, and affordable hardware and software are just some of the features that have led to their uptake in schools. However its value for learning has come with disagreements of its pedagogic value [462]. With a growing trend towards personalization of learning and increased flexibility and access, iPads are one such tool that can assist students to take ownership of their learning [463].

Furthermore, there is the realization of the Internet of Things (IoT) which depends mostly on the development and integration of key applied technologies, as well as their implementation and acceptance on a society level. The concept of (IoT) is about "*a world in which all electronic devices are networked and every object, whether it is physical or electronic, is electronically tagged with information pertinent to that object*" [464]. According to Zhang and Zhu [465] the four key technologies of IoT can be organized into: radio-frequency identification, sensor technologies, smart technologies and nanotechnology. The next step of the IoT vision is to interconnect people and objects over the Internet from "*any-time, any-place*" for "*any-one*" into "*any-time, any-place*" for "*any-thing*" thus, creating a smart environment, leading ultimately to a more convenient way of life for everyone [466]. A recent study [467] estimates that by the year 2020 there will be 50 billion objects that are connected to the Internet. It is expected that this movement will provide new business opportunities for hardware manufacturers and ICT companies at an unforeseen scale.

IoT devices typically contain sensors that can provide information about the environment they are located. Third party companies or organizations disseminate data produced by the devices through a wireless internet connection to cloud-based data storage services. In addition to storing the data, these services allow their users to analyze the data application programming interfaces (API) that allow creating solutions at ease, allowing very short development cycles.

It is clear that the emergence of the IoT will have a transformative effect on society and thus it requires on education. The NMC Horizon Report on Higher Education [468], mentioned the IoT for the first time in its 2012 edition and predicted its likely adoption by 2017. There is a need for an education provision that can empower the new generation of digital citizens who can understand both the technologies that underpin the IoT, as well as the societal impacts of widespread adoption of these technologies [469]. Moreover, the next generation of engineers should understand how to design and build technological systems that reflect the altered expectations of openness and participation. For CS the challenge is to develop new forms of scalable education that are able to accommodate the large numbers of students around the world, that are attractive to potential students with various interests and that deliver an innovative curriculum that reflects the radical changes in computing technology [469].

In response to these challenges the Open University in the UK has created a program to reform its CSE, by offering a new introductory course designed around IoT concepts [469]. The program aims to encourage undergraduate students to learn CS concepts like algorithms, programming and software design with IoT technology, rather than merely learning about the IoT.

A study by [470], described an experimental course where students develop IoT applications by using a) a Problem-BL approach to identify problems and come up with solutions for improving them and b) a Project-BL approach where these solution ideas were made real by creating projects. According to the authors the experience was very successful and taught students about technical aspects that were related to the IoT in a creative and collaborative way.

Emerging technologies will be essential for the foreseeable future are worthwhile for teaching CS principles. It is possible to incorporate the concept of the IoT early on CSE because it orients students towards the future of computing and society. Additionally, the physical computing aspects draws in students who would never have considered a traditional computer science course (according to direct testimony from our students and public discussions on Facebook etc.).

3.5 Summary

According to the thesis' objectives described in Chapter 1, the first step towards developing the framework for alternative teaching in CSE, is to discuss the issues that traditional teaching methods have. In this chapter, previous research endeavors in the area as well as their importance regarding student engagement and learning from alternative teaching methods are discussed. Every known method used in CSE is presented.

Part III - The Framework

Chapter 4

> *To pursue science is not to disparage the things of the spirit. In fact, to pursue science rightly is to furnish the framework on which the spirit may rise.*
>
> *Vannevar Bush*

Framework for Alternative Teaching

When teaching CS, many parameters need to be considered, like the students' age and experience or the learning goals. Learning different CS concepts more effectively, requires the development of a multi-level framework that could be applied in different contexts using different teaching approaches to a multitude of application domains. A framework that actively incorporates alternative teaching methods can cover diverse learning situations.

4.1 A new conceptual Framework

A conceptual framework defines a particular point of view within a discipline from which the researcher focuses his or her study [471]. This theoretical perspective identifies underlying assumptions, concepts, values and practices that constitute a way of viewing reality. Furthermore, every instructional design system is supported somehow by one or more learning theories and through them by ways of knowing or seeing different concepts (section 3.2). Although learning theories describe and try to explain how people learn, the main aim of instructional design is to provide guidance on the practical task of designing learning experiences [472].

In order to achieve that, this thesis proposes a framework for the effective implementation of ATM in CS. This new framework is called the ATMF (Alternative Teaching Methods Framework). ATMF is not a mechanism for evaluating researchers' approaches to instructional research. ATMF is a broad description of the context, characteristics, content and sequence of learning expected of all learners - but not at the level of detail of grade-by-grade standards or, at the course description and standards. As this thesis lays out, the framework is intended as a guide to instructors (formal or informal settings) as well as for curriculum designers, assessment developers, researchers and professionals responsible for CSE. Thus, it describes the major concepts, and disciplinary core ideas that all instructors should be familiar with in order to enhance their teaching with alternative strategies, and providing an outline of how these practices, concepts, and ideas should be developed across different settings.

There are several frameworks for research on instructional design (see Paas et al. [473] for an overview), and there is also rich support for designers of educational resources, with, for example, Beetham and Sharpe [474] conceptual model for designing learning objects. What is missing is a framework that would bring together teachers, researchers and designers into one space specifically focused on ATM. Therefore, the aim of the ATMF is to address this gap and to serve as a guide, or scaffold, for educational researchers entering the field of teaching and learning research in CSE.

The current literature and discussion on ATM that was summarized in the previous section (3.4), has been the theoretical framework on which the studies reported here have been searched for. The selection of the frameworks that are presented in this chapter aim to provide a selective but wide picture of what is available today and on how teaching is done and conceptualized. Furthermore, to match teaching methods with goals, the design of instructional materials should be grounded in a research-based framework [475]. The ATMF is based on empirical research as presented in the following sections.

4.2 The Framework' s Principles

The conception of a new framework in educational design research seeks to promote deep understanding of innovations and the factors that affect improvement in local contexts [476]. In that view, the ATMF does not purpose to offer guidance for measuring the efficiency of such engagement nor to offer universal solutions. Rather, the ATMF is a case within the larger framework of educational design-based research (e.g. [477]) and complements current efforts in the design-based research area which focus on process-oriented (e.g., [478]) or concept-oriented models (e.g., [479]).

Furthermore, the highly dynamic field of CS demands that learners must be empowered with reflective lifelong learning skills in order to be successful. They need to develop skills such as CT, problem solving, teamwork, communication, critical thinking, and creativity. The traditional teacher-centric pedagogy is focused on the course content and only on transferring knowledge to the students whereas a learner-centric view is focused on assisting students to develop or build knowledge [480, 481]. In CSE a move to a learner-centered design is recommended [66, 482], because the pedagogy varies from developing skills focusing on the understanding of the machine aspects like CT [67] or real world application aspects such as object-oriented design and development [483].

In order to achieve the above, the ATMF is built on a foundation of constructivist and collaborative learning theories [484-487] where knowledge is constructed: physically by active learning; symbolically by the creation of mental representations; socially by sharing understanding; and, theoretically by explaining things having incomplete understanding. The basic principles used to develop the ATMF are:

- Keep it simple: clarity and simplicity prioritized over a comprehensive list of CS topics.
- To go over old ground: based on past research in the area of CSE, and apply it to the teaching and learning process of various subjects.
- Research based: built on empirical research that is linked to relevant theory.

Moreover, about teaching and learning:

- Alternative teaching and learning: create a supportive and active learning environment to engage students by implementing a variety of ATM.
- Through learner's eyes: student understanding by investigating their beliefs and attitudes towards alternative learning.
- Through instructor's eyes: experimental research based on what teachers perceive about alternative teaching.

To ensure the validity of the framework, it was necessary to develop it in two consecutive research steps. Firstly, a theoretically derived conceptual framework was established. Secondly, empirical research was used to refine the framework.

Figure 23 illustrates the five phases of the methodology that was adapted in order to develop the ATMF:

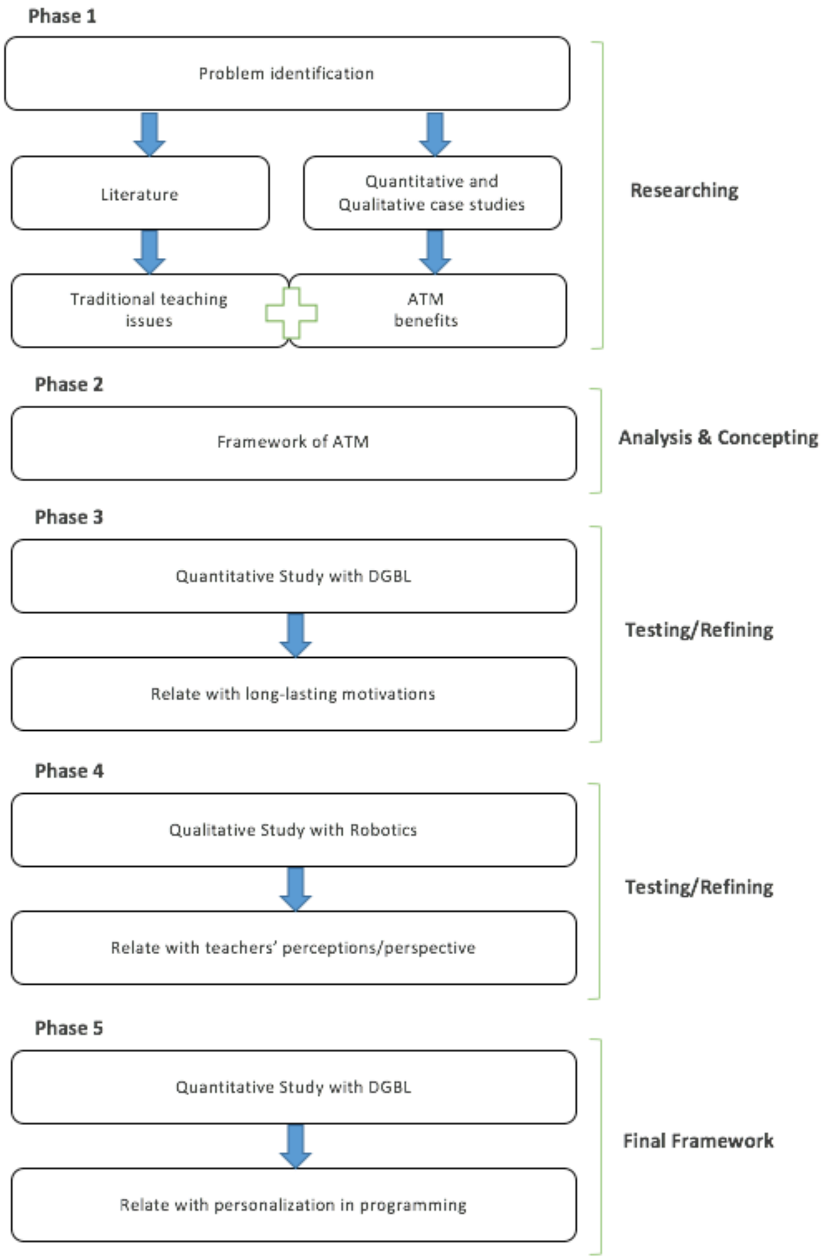


Figure 23. The five-phased methodology adapted for developing the ATM framework.

Phase 1: Research other frameworks and contact initial empirical study with quantitative and qualitative data. The process on one hand reveals the teaching issues that the traditional teaching methods have and on the other hand highlights the main advantages of the ATM.

Phase 2: Analysis of the concepts used at the ATMF. In this phase the framework is initially shaped through literature review and empirical research.

Phase 3: Test and refine the ATMF. In this phase the framework is tested and enhanced with additional elements.

Phase 4: Test and refine the ATMF. In this phase the framework is tested again and enhanced with additional elements.

Phase 5: Test and shape the final ATMF. In this phase the framework is tested, enhanced with additional elements and finalized.

4.3 Researching - Other Frameworks and Models

To assist in the development of the ATMF, background expert work was reused. In a first step, teaching methods definitions through research papers as well as curricula about CS and Computing were analyzed. Moreover, the frameworks and models that are presented in this thesis have been identified and collected in the following ways:

- Google Scholar searches of concepts related to teaching frameworks and/or models and/or taxonomies (e.g. teaching framework; learning framework; teaching model; learning model; teaching and learning taxonomy, CS framework, Computing framework, Science framework).
- Searches for the expression *teaching framework* and related terms (as above) in academic publication databases (ACM digital library, ERIC).
- Searching and browsing of curricula documents (primary, secondary education and higher education), on previous work carried out by CSTA, ACM and IEEE.

Many articles and books were identified as relevant. There were also found many useful websites, many of which are gateways to other sources. In this thesis are presented selected frameworks in terms of purposes and potential uses, by applying a consistent set of criteria. Each framework was dealt by taking into account the following aspects:

- Description and intended use (nature and function: framework/model/taxonomy map/list the domains and/or sub-domains addressed the principle or principles used in constructing the framework structural complexity and level of detail broad categories covered/teaching categories/stated purpose).
- Relevance for teachers and learning (actual and potential areas of application/implications for understanding/teaching and learning implications for practice/actual and potential use in research).

Therefore, the frameworks, models and taxonomies that are presented are the following:

- ✓ The Model of Educational Reconstruction.
- ✓ The Framework for Teaching.
- ✓ First Principles of Instruction.
- ✓ K12 Computer Science Framework.
- ✓ Bloom and SOLO taxonomies.
- ✓ Learning Technology System Architecture.

4.3.1 The Model of Educational Reconstruction

The Model of Educational Reconstruction (MER) was proposed for Science Education by Ulrich Kattmann and Michael [488]. The key idea of the model was that the science content of a teaching unit is not given, but has to be reconstructed according to the perspectives of the students as well as correspondent to the structure of the content, which was represented by a triangle of three equally important components (Figure 24). Teachers should perform the "*design of learning environments*" after iterating the two first steps "*investigation into students' perspectives*" and "*clarification and analysis of science content*", aiming to adopt subject matter knowledge as presented in textbooks or other scientific publications to the perspectives of the students in such a way that suitable teaching content could be constructed [489].

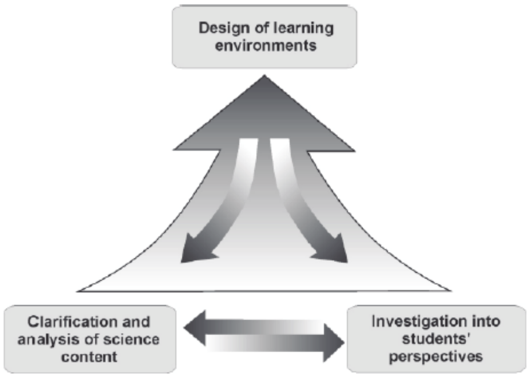


Figure 24. The Model of Educational Reconstruction as proposed by Ulrich Kattmann and Michael [488].

The original MER was enriched and refined by Duit [490]. The core element of this refined model is formed by the analysis of the content structure that comprises the subject matter clarification and the analysis of educational significance (Figure 25).

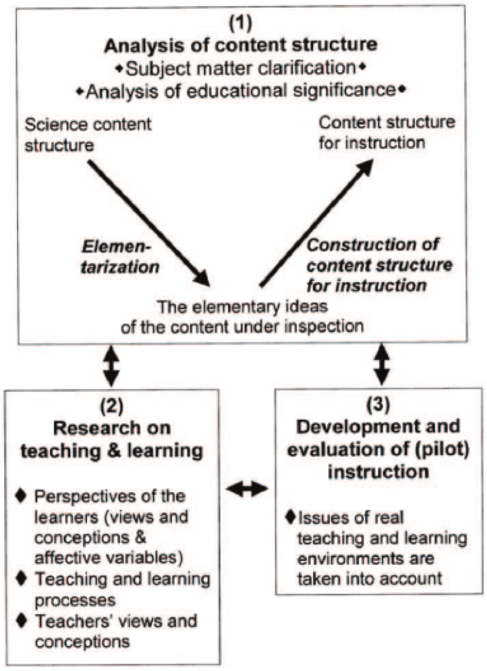


Figure 25. Educational Reconstruction according to Duit [490].

The refined model provides a theoretical framework for studies as to whether it is meaningful and possible to teach particular areas of science. It draws on the need to bring content related issues and educational issues into balance when teaching and learning are designed and aims at the improvement of understanding science [489]. The core element of this refined model is formed by the analysis of the content structure that comprises the subject matter clarification and the analysis of educational significance. It should be performed in two steps:

1. Elementarization, which results in the identification of the relevant elementary ideas of the content.
2. Construction of content structure for instruction.

The outcome of the second step is not only a simplification of the science content, but also an enrichment of this by putting it into contexts that make sense for the learners [490].

Furthermore, the MER adopts the idea of the fundamental interplay of all variables determining instruction by Heimann et al. [491] as presented in Figure 26. In this model the key interest is the students' learning process, while the *intentions (aims and objectives)* of instruction form the most significant frame for the process of designing instruction. Students' intellectual and attitudinal preconditions as well as socio-cultural preconditions significantly influence the interaction of the components shown in the first line of Figure 26 and they allow asking the four key questions that shape the process of instructional planning: *why*, *what*, *how* and *by what*.

Intentions (aims and objectives)	**Topic of instruction** (content)	**Methods** of instruction	**Media** used in instruction
Why	**What**	**How**	**By What**
Students' intellectual and attitudinal preconditions (e.g., pre-instructional conceptions, state of general thinking processes, interests and attitudes)		**Students' socio-cultural preconditions** (e.g., norms of society, influence of society and life on the student)	

Figure 26. On the fundamental interplay of instructional variables [491].

Following, Diethelm et al. [492] created a framework for the design and development of lessons in CS based on the MER [488]. The framework, illustrated at Figure 27, arranges four issues that are relevant for the design and arrangement of CS lessons and courses around the central sector of CS Phenomena:

- the clarification and analysis of the science content,
- the analysis of the social demands on the subject,
- the investigation of the students' perspectives on a topic,
- the investigation of the teachers' perspective.

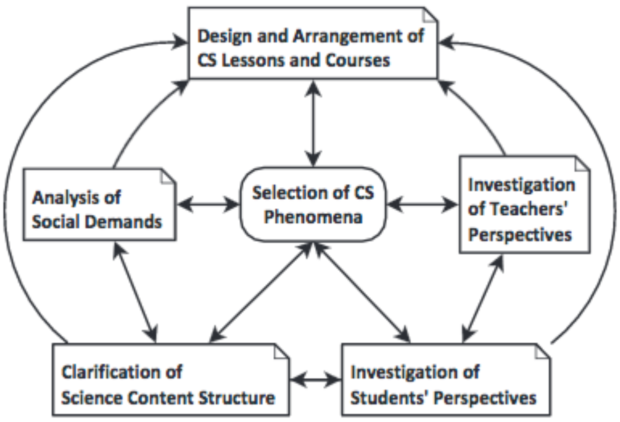


Figure 27. Educational Reconstruction for CSE by Diethelm et al. [492].

This framework is constructed according to specific goals, knowledge structures and teaching methods of CS at schools and is arranged around suitable "*real world phenomena*" in the sense of a teaching context, aiming to motivate the students, to open connections to prior knowledge or to show application situations of the intended knowledge. Thus, according to the authors [492] it may be applied as a road map for the "*CS in context*" approach as well as to other teaching approaches for CS at school.

4.3.2 The Framework for Teaching

CSE research is inevitably interdisciplinary. The circumstances of the classroom, the nature of education, and models of teaching and learning, are areas that are amenable to investigation only through humanitites. This means that we have to look to other disciplines for a theory base. The Framework for Teaching [30] is a research-based set of components of instruction, aligned to the InTASC[25] standards, and grounded in a constructivist view of learning and teaching. It is a general purpose Framework where the activity of teaching is divided into 22 components (and 76 smaller elements) clustered into four domains (Figure 28, left) of teaching responsibility:

- Planning and Preparation.
- Classroom Environment.
- Instruction.
- Professional Responsibilities.

The Framework for Teaching also includes many themes and ideas such as equity, cultural competence, high expectations, developmental appropriateness, accommodating individual needs, effective technology integration, and student assumption of responsibility (Figure 28, right).

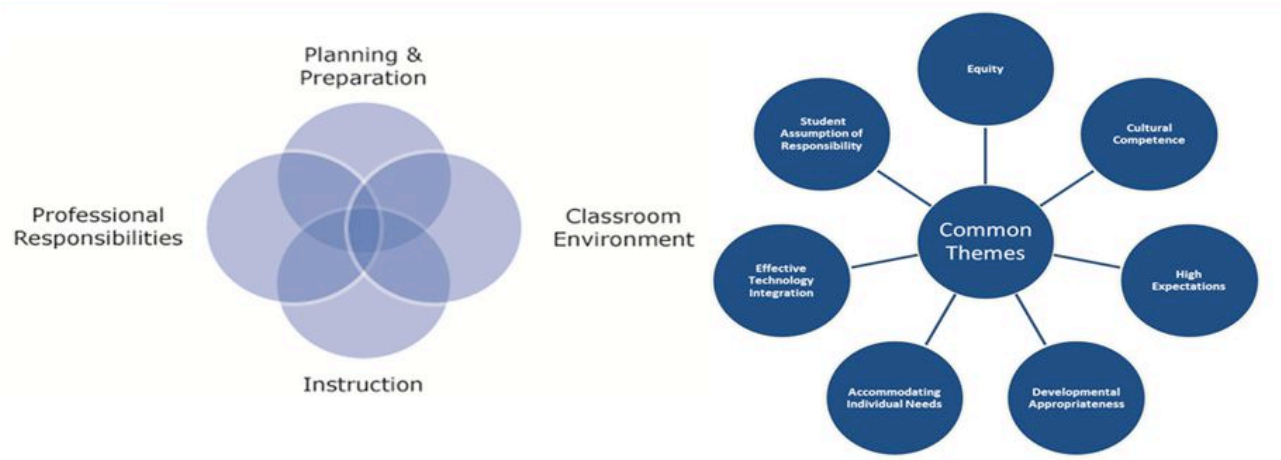


Figure 28. The Framework for Teaching [30].

In each one of the four domains, there are proposed standards and benchmarks at levels of performance called unsatisfactory, basic, proficient, and distinguished. The components in *Planning & Preparation* outline how a teacher organizes the content of what students are expected to learn, in other words, how the teacher designs instruction. The components in the *Classroom Environment* consist of the interactions that occur in a classroom that are non-instructional. The components in *Instruction* are what constitute the core of teaching, e.g. the engagement of students in learning context. Finally, the components in the last domain, about the *Professional Responsibilities* represent the wide range of a teacher's responsibilities outside the classroom.

As Danielson [30] states, the framework offers the profession of teaching a shard vocabulary as a way to communicate about excellence by laying out important components that constitute professional practice. The framework also serves to communicate to the larger community the array of competencies needed to be an effective teacher. Nevertheless, using a framework such as the Framework for Teaching can help teachers to reach individual learners, reflect upon their efforts, and receive guided peer feedback regarding their developing skills [493].

[25] Interstate Teacher Assessment and Support Consortium set of model core teaching standards for K-12 education in USA: www.ccsso.org/documents/2011/intasc_model_core_teaching_standards_2011.pdf

4.3.3 First Principles of Instruction

Based on an analysis of several instructional theories, models, and best practices, Merrill [494] proposed that effective teaching implements five fundamental principles in order to increase student learning. This framework is known as the "First Principles of Instruction" and comprises of five components describing when learning is promoted [494]:

- *The demonstration principle*: Learners observe a demonstration of the problem, from the instructor and/or peers and are guided to relate general information or the organizing structure to specific real world worked examples.
- *The application principle*: Learners apply their new knowledge by solving problems and receive feedback and coaching that is gradually withdrawn; application can include having students engage in peer-collaboration.
- *The activation principle*: Learners recall and activate prior knowledge and experience and therefore relevant cognitive structures by describing or demonstrating relevant, sharing previous experience with one another, and/or recalling or acquiring a structure for organizing new knowledge.
- *The integration principle*: Learners integrate their new knowledge into their everyday world by reflecting on, discussing, presenting, or defending.
- *The task-centered principle*: Learners engage in a task-centered instructional strategy based on a progression of whole real-world problems or tasks.

Merrill [494] converted these principles to a systematic cycle of instructional phases. The phases are based on a Problem and begin with Activation, followed by Demonstration, Application, and Integration (Figure 29).

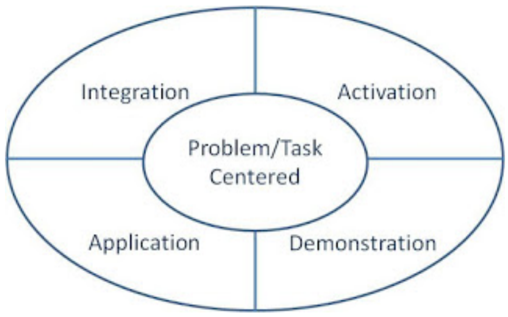


Figure 29. First Principles of Instruction by Merrill [494].

Merrill defines a principle as a basic method, and describes it as a relationship that is always true under appropriate conditions regardless of program or practice (variable methods) [494]. These principles constitute a set of fundamental elements common to all effective instructional design. Merrill [494] hypothesizes that:

1. Learning from a given instructional program will be facilitated in direct proportion to the implementation of first principles of instruction, and
2. learning from a given instructional program will be facilitated in direct proportion to the degree that first principles of instruction are explicitly implemented rather than haphazardly implemented.

The success of a given instructional program will be directly proportional to how well and how deliberately these principles are implemented [494].

Similarly, Keller [495] proposed the first principles of motivation that are common to all learning settings. In order to have motivated students [495]:

- ✓ their curiosity must be aroused and sustained,
- ✓ the instruction must be perceived to be relevant to personal values or instrumental to accomplishing desired goals,
- ✓ they must have the personal conviction that they will be able to succeed,
- ✓ and the consequences of the learning experience must be consistent with the personal incentives of the learner.

The First principles of Instruction are intended to be domain non-specific and so should be applicable to almost any discipline. Becker [496] in his research applied the principles defined above to CS instruction and concludes that, following these first principles will not only help students learn about CS concepts, but will also teach them how to be computer scientists. The principles in CSE may include engaging in real-world problems, using existing knowledge as the foundation, demonstrating new knowledge, and applying new knowledge, integrating it into the student knowledge base. In other words, challenge the students with new material that connects to their own by giving them examples that they can show to others.

The First Principles of Instruction can be a powerful framework for organizing and incorporating active learning strategies because it provides a clear framework in which active learning principles can be implemented [497].

4.3.4 The K-12 CS Framework

Recently there has been a major shift in the USA about how to make CS part of core academic work, responding to parent demand for their children to have access to CS[26]. CSTA, ACM, and Code.org joined forces with several advisors within the computing community (higher education faculty, researchers, informal educators, and many CS teachers), several states and large school districts, industry stakeholders, technology companies, and other organizations to steer a process to build a framework for K-12 education. The K-12 CS Framework is built upon the idea that CS provides foundational learning benefiting every child and gives students a set of essential knowledge and skills important for their learning and for their future careers and interests [4].

The framework identifies key CS concepts and practices that students exiting grades 2, 5, 8, and 12 (up to senior high school corresponding to the Greek educational system), expect to know. It gives guidance on how schools can engage in CS issues, approach problems in innovative ways and create computational artifacts with a personal, practical, or community purpose. CSTA has its own independent process for developing detailed CS standards [498] and thus the K-12 CS Framework avoids establishing standards.

The K–12 CS Framework presents the big ideas of CS through five core concepts and seven core practices (Figure 30). The concepts of the framework represent major content areas in the field of CS and are as following:

1. Computing Systems
2. Networks and the Internet

[26] Google Gallup 2015: Pioneering Results in the Blueprint of U.S. K-12 Computer Science Education http://csedu.gallup.com/home.aspx , last accessed March 2017

3. Data and Analysis
4. Algorithms and Programming
5. Impacts of Computing

Each of the core concepts incudes sub-concepts that coherent learning progressions that span kindergarten to grade 12. The selection of core concepts and sub concepts was informed by related work of the ACM Model K–12 Curriculum [11]; the CSTA 2011 standards [498]; the Advanced Placement CS Principles curriculum framework [37]; the Denning Institute's Great Principles of Computing [499]; and international frameworks, such as the United Kingdom's national computing program of study [500].

The core practices of the framework are:

1. Fostering an Inclusive Computing Culture
2. Collaborating Around Computing
3. Recognizing and Defining Computational Problems
4. Developing and Using Abstractions
5. Creating Computational Artifacts
6. Testing and Refining Computational Artifacts
7. Communicating About Computing

These practices represent the behaviors that computationally literate students use to fully engage with the core concepts of CS. The framework's learning progressions describe how students' conceptual understanding and practice of computer science grow more sophisticated over time. According to the authors, the concepts and practices are designed to be integrated to provide authentic, meaningful experiences for students engaging in CS [4].

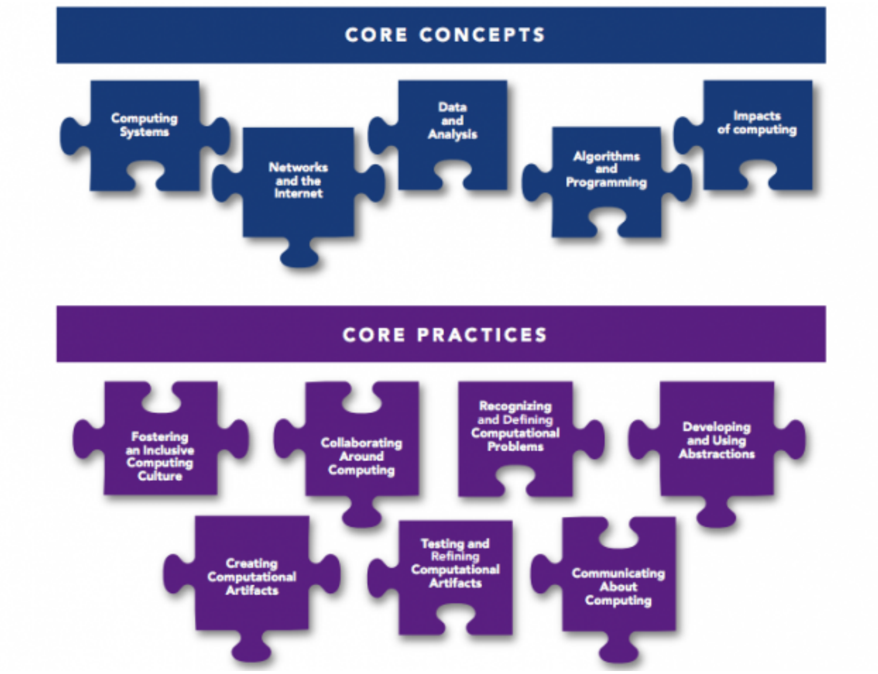


Figure 30. K-12 CS Framework core concepts and practices [4].

Furthermore, the concepts and practices of the K–12 CS Framework are not detailed lesson plans and activities in the form of curriculum. They are a high-level guide that can be used to inform the development of standards and curricula. As illustrated in Figure 31, the framework provides building blocks of concepts (that students should know) and practices (that students should do) which can be used to create standards (performance expectations of what students should know and do).

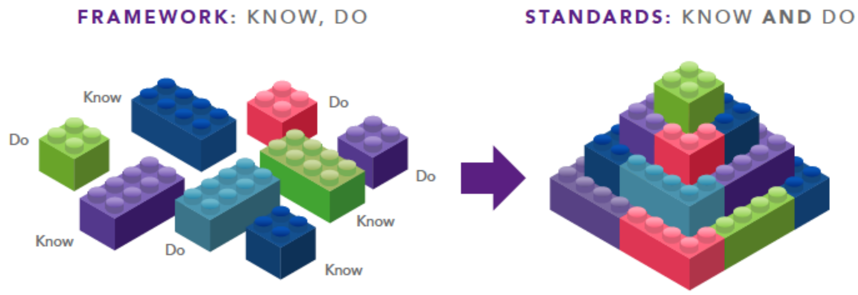


Figure 31. Building blocks for standards [4].

Moreover, the seven core practices of the K-12 CS Framework, describe the behaviors and ways of thinking that students in today's world should have. Figure 32 displays the practices that integrate each other and contain language that intentionally overlaps to illuminate the connections among them. The cyclical order from cyclical 1 to 7 suggests a process for developing computational artifacts which can follow many paths; in the framework, it begins with recognizing diverse users and valuing others' perspectives and ends with communicating the results to broad audiences (Figure 32). At the heart of the CS practices is CT and is defined by practices 3–6. Practices 1, 2, and 7 are independent, general practices in CS that complement CT. The thought processes involved in expressing solutions as computational steps or algorithms requires understanding the capabilities of computers, formulating problems to be addressed by a computer, and designing algorithms that a computer can execute [67].

Figure 32. The core practices of the K-12 CS Framework and CT [4].

The goal of the K-12 CS Framework has been to provide a high-level framework for CSE by identifying the core concepts and practices of CS and describing what those concepts and practices look like for students at various grade bands. The framework is designed for multiple audiences such as researchers, education designers and educators can use to develop their own set of standards, curriculum and instructional approaches in teaching CS. The community that has developed and supported this project believes that the K–12 CS Framework is an initial step to inform, inspire, and drive the implementation work required to make the vision of the framework a reality CS for all students [4].

4.3.5 Taxonomies

Some learning objectives are harder to achieve than others. For instance, it is much harder to learn to evaluate the quality of computer programs than to list programming keywords [501]. In the 1950s, a group of educators led by Benjamin Bloom defined a taxonomy that divides learning into three domains: the cognitive, affective, and psychomotor [105]. In the same work, they further detailed the cognitive branch of the taxonomy by presenting a hierarchy of learning objectives ranked according to their expected cognitive complexity. Bloom's work has received wide acclaim in education and remains highly influential.

The six levels of the original Bloom's taxonomy (from lowest to highest), are:

1. **Knowledge**: the student can recall specific facts or methods. This level is characterized by verbs such as enumerate, name, and define;
2. **Comprehension**: the student understands the meaning of facts or concepts. This level is characterized by verbs such as explain, discuss, and paraphrase;
3. **Application**: the student can solve problems by applying knowledge to new concrete situations. This level is characterized by verbs such as produce, implement, and solve;
4. **Analysis**: the student can break down information into its parts to determine motives or causes, or to make inferences. This level is characterized by verbs such as analyze, discriminate, and infer;
5. **Synthesis**: the student can combine elements in new ways to produce novel wholes. This level is characterized by verbs such as create, compose, and invent;
6. **Evaluation**: the student can make judgments about material in light of selected criteria. This level is characterized by verbs such as appraise, critique, and compare.

Many alternatives of the taxonomy have been proposed in the literature. An influential variant was defined by Anderson et al. [106]. This revised Bloom's taxonomy has two dimensions; a cognitive process dimension similar to that of the original taxonomy, and a knowledge dimension that specifies the type of content being processed – factual, conceptual, procedural, or metacognitive. Anderson et al. [106] also exchanged the places of the two last levels of the cognitive process dimension to produce a revised hierarchy: recall, understand, apply, analyze, evaluate, and create.

Many educators have used Bloom's taxonomy to motivate improvements to the instruction or assessment of CS concepts [107, 502-504]. The recent CER literature also features a thread that applies Bloom's taxonomy to study the learning of CS concepts and discuss the appropriateness of forms of assessment in introductory programming courses [505-508]. The revised Bloom's taxonomy is being used as a guideline by ACM and IEEE's work on developing their CS curriculum [509]. Furthermore, several variants of Bloom's taxonomy have been proposed to be particularly suitable for programming education [508, 510, 511].

Nevertheless, there is not a general agreement on precisely how to map the goals of CS and programming education onto Bloom's taxonomy. For instance, coding skills have been variously classified within the literature as the layer *understand* or *analyze*. Gluga et al. [512] reported that academics that used Bloom's taxonomy for the classification of programming assignments produced a variety of different classifications. CSE researchers have underlined the relevance of students' prior knowledge in determining the cognitive demands of an activity [108, 109, 513]. For instance, Thompson et al. [108] consider that applying programming knowledge involves solving familiar problems with new data or solving unfamiliar problems that match a familiar pattern or require an algorithm that is known to the learner.

Bloom's taxonomy is intended to be used as a tool for analyzing and designing courses and curricula. In particular, the taxonomy was created to emphasize that learning objectives that should not be set only at the lowest levels, as was being done in many traditional educational settings, but at all levels of the taxonomy. Krathwohl [514] a member of both Bloom's original group and the one that revised it decades later, looks back on how Bloom's taxonomy has been applied across disciplines:

"One of the most frequent uses of the original Taxonomy has been to classify curricular objectives and test items in order to show the breadth, or lack of breadth, of the objectives and items across the spectrum of categories. Almost always, these analyses have shown a heavy emphasis on objectives requiring only recognition or recall of information, objectives that fall in the Knowledge category. But it is objectives . . . in the categories from Comprehension to Synthesis that are usually considered the most important goals of education. Such analyses, therefore, have repeatedly provided a basis for moving curricula and tests toward objectives that would be classified in the more complex categories." p213.

In the field of computer programming, applying Bloom's taxonomy to course evaluation has not advanced the kind of shift that Krathwohl [514] describes, from knowledge towards more complex educational goals. Applying Bloom's taxonomy to introductory programming education has emphasized the fact that even introductory courses set the cognitive bar very high for future programmers. Researchers noted that to be capable of developing a (small) program which solves a given problem that has been expressed vaguely in non-programming terms, corresponds to synthesis or create levels of the taxonomy [107]. Typical introductory programming exams emphasize writing code, that goes to application and/or synthesis levels, depending on the question and the interpretation of Bloom [515-517]. Oliver et al. [518] examined computing courses in terms of a weighted average of the numbered Bloom levels of each type of assessment, named "*Bloom rating*". They discovered that programming courses consume high Bloom ratings, whereas more general courses on other CS topics had much lower ratings. The findings of another study [507] suggest that the higher up in the revised Bloom's taxonomy a programming task is, the more difficult it is for students succeed in it.

Nevertheless, some instructors have questioned the appropriateness of Bloom's taxonomy for the design of learning activities and assessments. This is the focus of the Structure of the Observed Learning Outcome (SOLO), a taxonomy formulated by from Biggs and Collis [519] empirical analyses of students' responses to learning tasks. SOLO's five levels can be used to categorize learner responses in terms of their structural complexity. Summarized, the levels of SOLO taxonomy are [520]:

1. ***Prestructural***: a response at this level misses the point or consists of empty phrases, which may be elaborate but show little evidence of actual learning.

2. ***Unistructural***: this kind of response meets only a single part of a given task or answers only one aspect of question. It misses other important attributes entirely.
3. ***Multistructural***: the response is 'a bunch of facts'. It expresses knowledge of various important aspects, but does not connect them except possibly on a surface level. The learner sees 'the trees' but not 'the forest'.
4. ***Relational***: the response relates and integrates facts into a larger whole that has a meaning of its own. It is no longer a list of details; rather, facts are used by the learner to make a point.
5. ***Extended abstract***: a response at this level goes beyond what is given and applies it to a broader domain.

SOLO describes a systematic progression in performance as an individual learns. First, from the prestructural through to the multistructural level, the learner makes quantitative progress, increasing the amount of knowledge they have. Progressing to the relational and extended abstract levels involves a qualitative change as meaning emerges from the increasingly well perceived connections between elements of knowledge (Figure 33). SOLO is intended to be used by teachers both for analyzing responses to learning activities (e.g., answers to questions) and for setting learning objectives. SOLO and Bloom's taxonomy have somewhat different perspectives – Bloom classifies learning objectives (skills), while SOLO classifies learning outcomes (responses to activities) – and they are, to a certain extent, complementary. Both can be used to characterize the objectives set for learners.

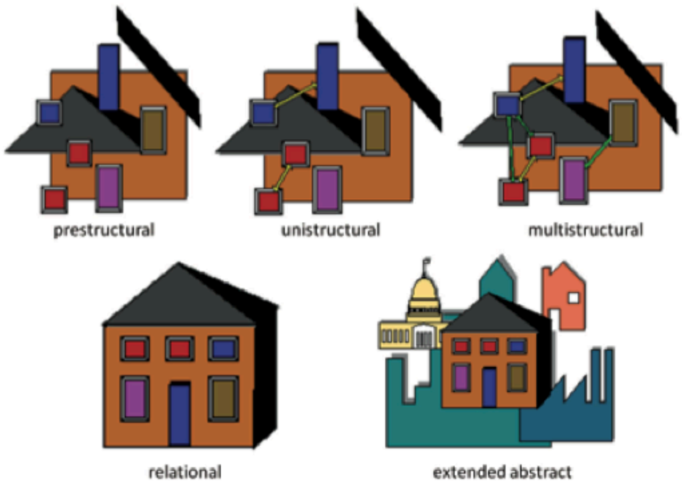


Figure 33. Metaphorical illustration of SOLO's five levels [521] .

Moreover, SOLO has been used to analyze programming assignments. Jimoyiannis [522] in his study provided information about students' mental models and representations of the programming variable and the assignment statement. Some of the difficulties reported are intrinsic or inherent to programming but there are also themes for redesigning instruction. The author concludes that traditional teaching approaches appeared to be inefficient to support students' developing of appropriate mental models of the basic programming concepts [522]. Related, a research project called BRACElet has applied SOLO to reading and writing coding tasks in the context of introductory programming [523-526]. The project presents an interpretation of how SOLO applies to simple code comprehension problems of the form "*explain what the following segment of Java code does",* and characterizes the four main levels as follows [525]:

1. Prestructural: substantially lacks knowledge of programming constructs or is unrelated to the question.
2. Unistructural: a description of one part of the code.
3. Multistructural: a line-by-line description of all the code (the 'trees').
4. Relational: a summary of what the code does in terms of its purpose (the 'forest').

Research suggests that the degree of structured responses to a code-reading task measured on such a SOLO-based scale correlates significantly and positively with their ability to write program code [524]. As for writing code, one suggestion for categorizing responses was drawn at the annual BRACElet workshop at AUT University [527] which characterizes the four levels as follows:

1. Prestructural: inability to write correct code;
2. Unistructural: ability to write a single small piece of code, e.g., to increment the value of a variable.
3. Multistructural: ability to write combine a few statements to write a multi-line solution based on a detailed specification or pseudocode. E.g., completing a method so that it returns false if a given book is on loan but true otherwise.
4. Relational: ability to write code to solve a problem which has not been specified to the extent that the problem represents pseudocode for the solution. E.g., writing a class to represent library books.

SOLO taxonomy, requires an understanding of programs that reaches the relational level for successfully writing programs. Program design tasks may even require students to transfer programming concepts beyond what they have encountered or learned about and to the extended abstract level [522]. Brabrand and Dahl [528] analyzed stated requirements of CS, mathematics, and natural science courses of a Danish university where all teachers are required to use the SOLO taxonomy as they specify course goals. The authors found that CS courses in general had considerably higher SOLO levels than natural science courses and (even more clearly) than mathematics courses. Typical programming related competencies desired were relational at a high level.

Both Bloom's taxonomy and SOLO illuminate that in CSE the learning objectives are cognitively challenging, the expected learning outcomes are structurally complex. Programming is a key skill within CS, and is an ever more important tool for non-computer-scientists. It seems that CS instructors may be systemically underestimating the cognitive difficulty in their instruments for assessing programming skills of novice programmers [507]. The level of difficulty of programming assessments, whether or not inherent in the subject itself, presents a significant and possibly unfair barrier to student success.

4.3.6 Learning Technology System Architecture

The influence of technology in the learning process is very important. One aspect that describes the intellectual and technical development of educational technology is the theory and practice of educational approaches to learning [529]. Up to the time this thesis was written, a number of models and frameworks have been used to examine the adoption and integration of technologies into the classroom. These include such models as the TPACK framework [530]; the Replacement, Amplification and Transformation (RAT) framework [531]; The Technology Integration Matrix

(TIM) [532]; the framework of Substitution, Augmentation, Modification and Redefinition (SAMR) [533] and the Learning Technology System Architecture (LTSA) [534].

The Institute for Electrical and Electronic Engineers[27] (IEEE) consumes the Learning Technology Standards Committee (LTSC). The LTSC has created many standards that most groups working in creating specifications at the education field are based on. The LTSA that was launched in 1999 is a an architecture that become standard for many learning technologies and covers a wide range of systems in education [534]. The model' s specifications are to:

- ✓ provide a framework for understanding existing and future systems,
- ✓ promote interoperability and portability by identifying critical system interfaces,
- ✓ incorporate many years while remaining adaptable to new technologies and learning technology systems.

The LTSA considers teachers and students as entities to a system that delivers specific processes to learning and assessment by using resources and records stored in databases as shown at Figure 34.

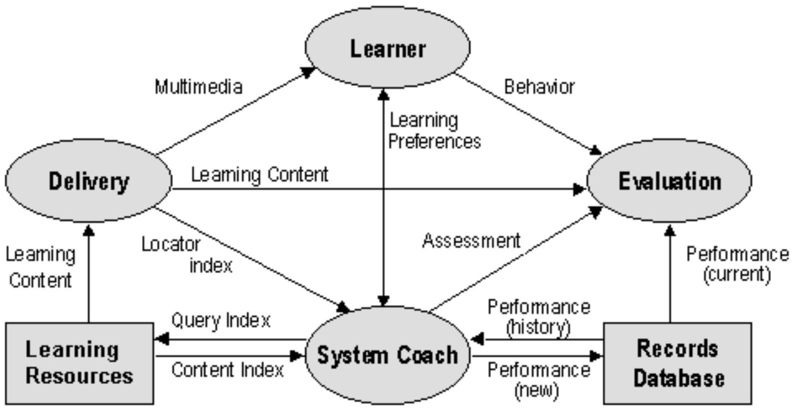


Figure 34. Learning Technology Systems Architecture (LTSA) by IEEE 1484 [534].

This architecture defines the basic entities which are found in technology learning systems (e.g. e-learning): learners, teachers, learning contents and assessments. In the whole process there is an element always present, the content. Content needs to be maintained, delivered, fixed and could receive many other operations along their use through learning systems. As many of these operations can be simplified or aided by software tools, it is absolutely necessary to use a standard format in the representation of the contents. These learning contents are known as *learning objects*, which can be defined as minimal units in which training materials can be organized, in order to ease the management of content: creation, indexation, storage, delivery, use, reuse, evaluation and training improvement [535]. Finally, they can be later combined in order to create greater units of instructions (e.g. lessons and courses).

4.4 Analysis - Concepting

Most frameworks/models described above were developed for general purposes or science education, where it is self-evident to arrange lessons around experiments and phenomena. It is also accepted that physics, chemistry and biology are taught in compulsory courses for over a century

[27] IEEE: http://www.ieee.org/ , last accessed February 2017

now. Additionally, there is a strong logic about the core content of the curricula of these subjects. Instructors are educated according to a long tradition and to curricula that are at least similar among different countries. For CSE not all these apply. Therefore, we have to take a closer look at the discussed frameworks in order to adopt them for CS. Several issues that are particularly important for CSE are missing, respectively not represented in a way that meets this importance. Hence, we present a specific framework for CSE that was derived by explication, rearrangement and extension from other frameworks, models and taxonomies.

The ATMF comprises the need to consider context, content, pedagogy, instructor and the learner as part of the design process. In particular, in advance of planning a lesson with the use of ATMF, it is suggested that a comprehensive learning analysis be produced that sufficiently covers the following standpoints:

Dimension 1: Context

- Component 1a: Education Level
- Component 1b: Environment (physical)
- Component 1c: Educational system

Dimension 2: Participants' Characteristics

- Component 2a: Learner related
- Component 2b: Instructor related

Dimension 3: Content

- Component 3a: Learning
- Component 3b: Teaching

Dimension 4: Evaluation

- Component 4a: Learner related
- Component 4b: Instructor related

Each component defines a distinct aspect of a dimension; components are consisted of elements that describe specific features. For example, Dimension 4, Evaluation, contains two components. Component 4a is Learner related, which consists of five elements: Knowledge/Learning Performance, Motivation, Skills, Metacognition and Beliefs.

Dimension 1. Context: The components in Dimension 1 describe the surrounding environment around the learning process (Figure 35). The first dimension deals with the need to consider the educational level and the place where learning is taking place, the resources available (e.g. access to laptops/computers, mobiles, technical support), and the disciplinary context (e.g. in school, in ICT lab, in a university, at home, in the workplace). The components establish the environment that supports the learning and are not directly associated with the learning of any particular content, instead, they set the stage for all learning processes. The specific elements of the learning environment are captured in three categories: Teaching Level, Classroom Level and Educational System. The Education Level includes the type of education where teaching/learning occurs. If it is about formal education that it must include the type of school (e.g. vocational education) and the level of education (e.g. university).

Dimension 1: Context		
Component **1a**	**Education Level**	
	Element 1	Formal Education / Type of School / Level
	Element 2	Informal Education
	Element 3	Non-formal Education
Component **1b**	**Environment (physical)**	
	Element 1	Infrastructure (availability)
	Element 2	Supporting Resources
Component **1c**	**Educational system**	
	Element 1	Policies
	Element 2	Compulsory Curriculum
	Element 3	CS standards
	Element 4	General public attitudes
	Element 5	Time restrictions

Figure 35. Components end Elements of Dimension 1.

Moreover, the context of teaching includes anything in the surrounding environment that influences teaching and learning. The physical environment includes the classroom where teaching/learning occurs including ICT infrastructure and other supporting resources. For instance, the arrangement of the desks encourages some kinds of interactions and discourages others. Other factors such as robotics can make a difference.

Finally, when designing/planning a lesson, one must take into account restrictions from the educational system that learning occurs. Component 1c is correlated with formal education settings. Elements 1 to 3 are relate to restrictions that occur from the principles and government policy-making in the educational sphere as well as the collection of laws and rules that govern the operation of an education system. Since this framework is about ATM, often it is important to take into account the general public's attitudes [536] about alternative teaching (Element 4). Lastly, time seems to be an important factor when dealing with ATM (see Research Study 3: Game Based Learning).

Eventually, the culture of where the teaching and learning take place determines what is valued, rewarded and recognized in the context. All these information is critical to matching the best solution to support effective teaching.

Dimension 2. Participants: This dimension describes the participants' characteristics in two main components, learners and instructors (Figure 36). The environment including the relationship between instructor and learners and the cultural norms (characteristics) play a significant role in what can and does occur during the teaching and learning. The concept of learner characteristics is used in the sciences of learning and cognition to designate a target group of learners and define those aspects of their personal, academic, social or cognitive self that may influence how and what they learn [537]. Both learner and instructor characteristics are important for instructional designers as they allow them to design and create tailored instructions for a target group.

Component 2a, about learner related characteristics comprises of eight elements that describe the personal, background, social/emotional and personality features. Personal characteristics relate to demographic information such as age, gender (elements 1 and 2). Next, background characteristics relate to cultural differences, social economic status (elements 3 and 4), and prior experience of a learner group such as ICT use (elements 5 and 6) and attitudes towards alternative learning (element 7). Finally, about component 2a, the learners' personality characteristics are considered very important. Examples of personality traits are psychological preferences in how people perceive the world and make decisions and the ability to understand and interact effectively with others. For instance, cognitive style seems to be an important factor in learning programming through games (see Research Study 5: Personal Learning Characteristics).

Similarly, dimension 2, includes component 2b about instructors' characteristics that can have importance in the teaching process (Figure 36). Again, the personal demographic information about age and gender are the first two elements. The next element 3 is about teaching experience, a very important factor in addressing difficulties during teaching [538]. Research [539] has shown that compared to a teacher with no experience, the benefits of experience rise monotonically after 21-27 years of experience, with more than half of the gain occurring during the first couple of years of teaching. Academic characteristics (element 4) are more education and background learning related such as discipline, prior knowledge, educational type, educational level and other qualifications. CS is relatively new and most instructors do not have a clear CS background. This might be an influencing factor when using alternative teaching as in Research Study 1: Peer Learning. Element 5 relates to skills and abilities that may support alternative approaches in teaching, like organization, critical thinking, passion, flexibility and enthusiasm.

Dimension 2: Participants' Characteristics		
Component **2a**	**Learner related**	
	Element 1	Age
	Element 2	Gender
	Element 3	Cultural background
	Element 4	Socioeconomic status
	Element 5	ICT use
	Element 6	ICT experience
	Element 7	Attitudes
	Element 8	Personality traits
Component **2b**	**Instructor related**	
	Element 1	Age
	Element 2	Gender
	Element 3	Teaching experience
	Element 4	Academic characteristics
	Element 5	Skills - Abilities
	Element 6	Attitudes
	Element 7	Personality Traits

Figure 36. Components end Elements of Dimension 2.

Component 2b also contains instructors' attitudes (element 6) about alternative ways of teaching. For instance, teachers that prefer traditional teaching believe that it gives better learning results and hesitate to include active-learning methods in their teaching, although they see some advantages in them (see Research Study 1: Peer Learning). Finally, personality traits (element 7) relate to such things as actions, attitudes and behaviors that determine different personality styles like cognitive style and intelligence type. Being positive and upbeat can influence learners around an instructor, and so can negativity. The personalities of both instructor and learner interacting with one another and with the content create a unique environment. It is expected that by taking into account the participants' characteristics can be designed and developed more efficient, effective and motivating instructional materials.

Dimension 3. Content: The components of the Dimension 3 reflect process workflows from teaching and learning theories (sections 3.1 and 3.2) and from the alternative teaching methods as described in section 3.4. Those components describe how an instructor organizes the content that the students are to learn and how he/she designs instruction (Figure 37).

Dimension 3: Content		
Component **3a**		**Learning**
	Element 1	Subject
	Element 2	Objectives
	Element 3	Short-term & Long-term Planning
	Element 4	Expectations for learning
	Element 5	Relations to other subjects / disciplines
Component **3b**		**Teaching**
	Element 1	Lecture-based
	Element 2	ATMs (see Figure 33)

Figure 37. Components end Elements of Dimension 3.

Component 3a comprises of five elements, related to what is the content to be taught. Elements 1 to 3, refer to lesson planning and effort to answer queries like: What is the subject to be taught? What are the objectives and goals that instructor aims to achieve? Does the lesson planning refer to short or long-term features? etc.

Instructors must have a deep and flexible understanding of their content areas and must be able to draw upon content knowledge as they work with learners to access information, apply knowledge in real world settings, and address meaningful issues to assure learner mastery of the content. In addition, instructors should make content knowledge accessible to learners by using multiple methods of teaching (Figure 38).

ATMs		
Element 2		**Teaching**
	Section 1	Peer Learning
	Section 2	Problem-Based Learning
	Section 3	Project-Based Learning
	Section 4	Studio-Based Learning
	Section 5	Inquiry-Based Learning
	Section 6	POGIL
	Section 7	Team Learning
	Section 8	Game-Based Learning
	Section 9	Educational Robotics
	Section 10	Non-Textual Programming
	Section 11	Contextualized Learning
	Section 12	CS Unplugged
	Section 13	Subgoal Learning
	Section 14	Programming Puzzles
	Section 15	Extreme Programming
	Section 16	Program Visualization
	Section 17	Competency-Based Learning
	Section 18	Social Networks Learning
	Section 19	Emerging Technologies

Figure 38. Alternative Teaching Methods (Sections of Element 2, Component 3a, Dimension 3).

Planning focuses on using a variety of appropriate teaching strategies to address diverse ways of learning, to incorporate new technologies to maximize and individualize learning, and to allow learners to take charge of their own learning and do it in creative ways. The instructor understands and uses a variety of ATM to motivate learners, to encourage them to develop deep understanding of content areas and their connections, and to build skills to apply knowledge in meaningful ways.

Furthermore, the instructor understands the central concepts, tools of inquiry, and structures of CS and creates learning experiences that make the discipline accessible and meaningful for learners to assure mastery of the content. It is expected that the lesson is planned in such a way that supports every student in meeting rigorous learning goals by drawing upon knowledge of content areas, curriculum, cross-disciplinary skills, and pedagogy, as well as knowledge of learners and the community context. With the use of appropriate teaching approach, the instructor can integrate skills like critical thinking, problem solving, creativity and communication, to help learners use content to propose solutions, forge new understandings, solve problems, and imagine possibilities.

Dimension 4. Evaluation: This dimension addresses the outcomes of the teaching and learning processes, which deal with assessment and evaluation. The outcomes (direct or indirect) are examined in two main components about learners and instructors (Figure 39).

Dimension 4: Evaluation		
Component **4a**	**Learner related**	
	Element 1	Knowledge / Performance
	Element 2	Motivation
	Element 3	Skills
	Element 4	Metacognition
	Element 5	Beliefs
Component **4b**	**Instructor related**	
	Element 1	Assessment Criteria / Method
	Element 2	Monitoring of Learning (Learning Analytics)
	Element 3	Feedback to Learners
	Element 4	Long-term Evaluation
	Element 5	Teaching Evaluation

Figure 39. Components end Elements of Dimension 4.

Component 4a comprises of five elements that relate to learners' assessment. Assessment is the systematic collection of information about learning, using the time, knowledge, expertise, and resources available, in order to inform decisions that affect student learning [540]. Element 1 relates to student learning during the learning process (formative) and after it (summative) and is measurable. Instructor may gather feedback to identify areas where learners are struggling so that she can adjust the teaching and learners can adjust the studying. Next, element 2 is about motivation and is related indirectly with the fist element. Motivation energizes, directs, and sustains behavior and conclusively affects learning [541]. Learners with high motivation reflect in personal investment and in cognitive, emotional, and behavioral engagement in learning activities and succeed better performance [542]. It is expected that motivation is highly affected by the alternative teaching and learners enjoy the learning process (see Research Studies 1 to 5, chapters 5 to 9). By implementing ATM, many valuable skills are developed like critical thinking, creative thinking, communicating, and collaborating (element 3). Such skills help learning (the measurable outcome) but they are also essential for success in school and beyond (the non-measurable outcome). Element 4 is about metacognition, which is "*cognition about cognition*", "*thinking about thinking*", or "*knowing about knowing*" and higher order thinking skills [543]. Metacognition can take many forms and it includes knowledge about when and how to use particular strategies for learning or for problem solving.

Effective instructional practice requires that instructors understand and integrate assessment, planning, like instructional strategies in coordinated and engaging ways (element 1). Beginning with the goals, instructors first must identify learning objectives and content standards and align assessments to those objectives. They must understand how to design, implement and interpret results from a range of formative and summative assessments including peer assessment. The multiple methods of assessment must be used to monitor learners' progress (element 2). In other words, instructors monitor learning by gathering information about each student's understanding and then using the data to adjust their instruction to effectively increase student learning. To this end, learning analytics may help instructors to identify difficulties that learners encounter, detect gaps and differentiate instruction, analyze which lessons resonate best and for who and why, customize material for learners and control the teaching. Learning analytics is a process of gathering and analyzing details of individual learner interactions in learning activities aimed at learner profiling [544]. The measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs can help teaching and learning [545]. In addition, instructor must engage learners in their own growth and guide their decision-making. Therefore, they can improve the learning through timely and detailed feedback (element 3). This knowledge is integrated into instructional practice so that instructors have access to information (element 4) that can be used to provide immediate feedback (element 3) to reinforce student learning and to modify instruction. Finally, element 5 relates to teaching evaluation. Is the teaching method that was followed the only acceptable teaching practice? Is alternative teaching an acceptable practice and are instructors encouraged to take risks? Are learners encouraged to take an active role in their own education? The answers to those questions may determine whether teaching is successful or not. Instructors that understand expectations and components of the evaluation process can become more aware of their teaching and of the learners' progress.

4.5 Shaping the Framework

Figure 40 provides the visual description that was used to for the ATMF.

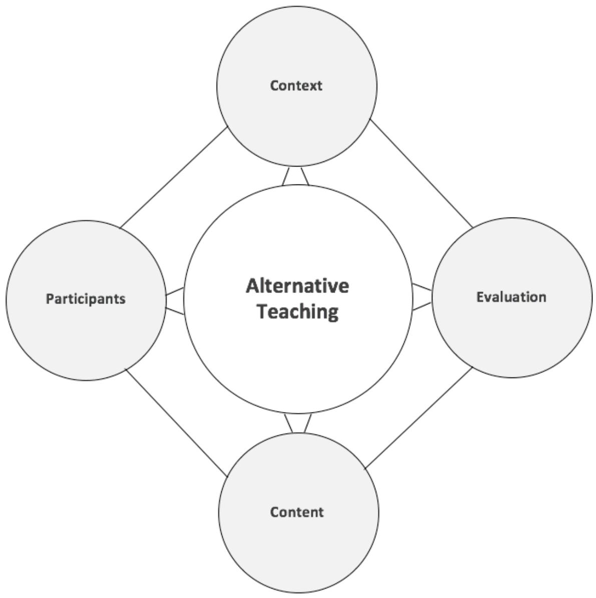


Figure 40. The ATM Framework for CSE.

This visual recognized the importance of multiple dimensions of: context (describing the setting, curriculum, policies and infrastructure), participants (describing the instructors and learners' characteristics), content (describing the subject matter, purposes and values, pedagogy and strategies) and evaluation (describing the outcomes for both learners and instructors).

Those four dimensions are presented as being a complex and interconnected whole with ATM at the hub connecting all the dimensions. ATM is in the center of all the process in order to emphasize that instructors have to think about and reflect upon the multiple dimensions as they investigated each topic or assignment of implementing ATM. The overall goal is to guide instructors in developing an integrated, interconnected knowledge for implementing effectively ATM in their teaching.

4.6 Summary

In this chapter, the proposed framework for alternative teaching within CSE was presented. Subsequently, previous frameworks, models and taxonomies were discussed and their potential was revealed. In addition, analysis of the framework's components and the shaping scheme that was used, were presented. The next chapter introduces the first experiment of ATM by following PL and CL methods in order to produce components that were used in the proposed framework.

Chapter 5

If we teach today's students as we taught yesterday's, we rob them of tomorrow.
John Dewey

Research Study 1: Peer Learning

In this chapter[28] we introduce the first stage of experiments on ATMs. In this research study (RS1), the students of a Greek junior high school collaborated to prepare the teaching material of a theoretical CS course and then shared their understanding with other students. The study investigates two alternative teaching methods (collaborative learning and peer tutoring) and compares the learning results to the traditional learning context. A test was used to measure all participating students' learning results and a questionnaire was distributed to record participant student attitudes towards the alternative teaching conditions. The questionnaire was designed to evaluate each aspect in terms of perceived knowledge, experience, satisfaction, diversity, oddness and interest. The analysis explores potential differences of students' learning results between alternative and traditional teaching and also differences in the two aspects in relation to students' preferences. Results provide evidence that active-learning methods can promote positive attitudinal shifts and improve skills in creativity, teamwork, collaboration and communication. Students perceived higher levels of learning than with traditional teaching. Finally, in terms of students' preferences, the majority wanted to have more courses taught with active-learning methods.

This chapter is organized as follows. In Section 5.1, we present background of active learning in CSE and aspects of the Greek educational system. In Section 5.2, we discuss the methodology that was adapted in this study. Then in Section 5.3, the results of the three experiments that were conducted are presented, while Section 5.4 discusses aspects of the alternative ways of teaching. Subsequently, the most important conclusions of this research study are summarized in Section 5.5. Finally, section 5.6 presents the elements that this study enhanced during the design of the ATMF.

5.1 Introduction

For this study an introductory theoretical lesson for the first grade of a Greek junior high school was used with three different experimental conditions consisting of two active learning methods and one traditional. Active learning methods "involve students in doing things and thinking about the things they are doing" [546]. In traditional lecturing, the amount of information retained by students, declines substantially after ten minutes [135]. Instead, when students participate in the learning process and collaborate in activities their motivation is increased and they develop high order thinking skills [546]. In classes with this kind of teaching, one can observe higher levels of energy and participation and, above all, effective learning [124, 547]. This chapter considers the terms "Alternative teaching" and "Active learning" as interdepended, in the sense that Alternative teaching is the one in which active learning occurs. Grissom [548] in the introduction of ACM's "*Special Issue on Alternatives to Lecture in the Computer Science Classroom*" attempts to raise awareness of evidence-based teaching practices that are effective alternatives to lecture. These practices include, among others, CL,

[28] Parts of this chapter have been published in: Theodoropoulos, A., Antoniou, A. & Lepouras, G. Educ Inf Technol (2016) 21: 373. doi:10.1007/s10639-014-9327-7

PI, POGIL and SBL as described in chapter 3. The author highlights that it is not sufficient that CS teachers simply become aware of these practices but also implement them in their teaching.

The multi-dimensional skills required from students nowadays raise challenges of how best to introduce CSE. In addition, educators are also searching ways to creatively integrate 21st century skills into the learning process, such as critical thinking, problem-solving, communication, collaboration, creativity and innovation [549]. It has been found that the application of alternative teaching may enhance essential 21st century skills among students [550, 551]. The traditional teaching approach where teachers give information to passive students appears outdated, and methods requiring actively participating students are worth researching [552]. Wilson et al. [9] examined the implementation of ACM K-12 standards [56], which include collaboration, in the United States of America. The authors point out that 21st century CSE should focus on developing deeper concepts and capabilities such as creation of new knowledge, innovation and imagination.

From the literature review, it seems that most ATMs have been applied and tested in higher educational settings. Few studies have tested these methods in secondary education (grades 7-12). In this study, the effects of active-learning methods are investigated together with student and teacher perceptions in a Greek high school. To the authors' knowledge, this is one of the first times alternative teaching methods are tried in a Greek high school. Active learning methods are still largely under-exploited in Greek schools and in addition, the traditional structures of the society (especially regarding education) imply that both students and teachers have specific expectations from a teaching session [553]. As Baran [554] explains, alternative methods in traditional educational settings are worth investigating since apart from their educational value, they can also contribute in the altering of traditional patterns of teaching and learning. The present study becomes particularly relevant, since currently the Greek educational system undergoes a transition phase and various teaching methods are explored [42].

Research in the field of education shows that ATMs, such as CL, PT and PA may considerably improve learning outcomes [546, 547, 555-557]. In this study we have researched these alternative and collaborative learning activities. CL (described widely in section 3.4) is an educational approach that involves small groups of students working together toward a common goal [558]. According to Gerlach [559], "*Collaborative learning is based on the idea that learning is a naturally social act in which the participants talk among themselves. It is through the talk that learning occurs*" (p. 12). During collaborative learning, students take almost full responsibility for working and building knowledge together. It has been researched that collaboration can facilitate learning [560-562]. Students retain the information for a longer time period and gain high-order skills such as analysis, creation and problem solving [563]. A study by Terenzini et al. [564] compared collaborative learning methods with traditional learning using 480 undergraduate students. Their results indicate that collaboration methods produce statistically significant and substantially greater gains in learning than those associated with traditional instructional methods.

In addition, students in the present study were also involved in PT (described widely in section 3.4). During PT students are responsible for their peer's learning. They are expected to acquire new knowledge and apply in class by teaching their peers [565]. It is a tutoring method where students teach other students. All participants learn more and demonstrate mastery when they are able to comprehensively teach a subject [566]. Furthermore, a student can form examples and relate to her peers on an entirely different level than an adult educator. Research has shown that explaining something to oneself and to others can promote learning [567].

Finally, in the present study, the aspect of PA (described widely in section 3.4) was also explored. PA, is the evaluation of work by one or more students who use their knowledge and skills in order to access their peer's work [568]. The main idea of PA is based on the concept that a group of different students will usually find more weaknesses and errors in a student's work and will be able to evaluate it more impartial than the student responsible for creating it [569, 570]. Past research in PA has shown the educational value of such practices [571-573].

Unlike other studies [574-576] that go beyond the surface of the terms CL and PL, for the time restrictions experienced in the current work, researchers only used generic definitions (as the ones presented above). The method developed and used, uses basic characteristics of the two terms. Accepting a tradeoff between time efficiency and depth of results gathered, the present study managed to collect valuable data from a traditional schooling system, applying new alternative teaching and learning methods in a restricted time setting (due to school's curricula). The initial results gathered (and presented below), provide an important first insight into the Greek Educational System and in particular to the teaching and learning of computing, becoming thus the basis for future work. Therefore, elements of CL and PL will be combined and presented in the methodology section.

5.1.1 CS in the Greek Secondary Education

This section aims to give an overview of the current situation in the field of CS in Greek Secondary Education. Secondary Education in Greece consists of two cycles: The Compulsory Secondary Education which is provided by Junior High School (Gymnasium) and the Post-Compulsory Secondary Education which is provided by High School (Lyceum) or Technical Vocational Schools. Students at Gymnasium are aged from 12 to 15 and for example the 1st grade corresponds to 7th grade of the K-12 educational system which is widely used in the United States of America.

The CS discipline was introduced in the curriculum of Greek Secondary Education gradually since the 80's[29]. Initially, during 1983-1992 it was introduced only in the Technical Vocational and Multidisciplinary Education. After 1992 the introductory course of Informatics was established in Junior High School. During 1992-1998 the new specialty for CS teachers was created. In school year 1998-1999 CS entered the High School curricula and was also reformed in the Technical Vocational Schools.

Today, almost all secondary schools are equipped with computer laboratories. However, many labs, especially in rural areas, have several shortcomings, especially concerning the quality of computers. The technological infrastructure of schools is supported by the Greek School Network[30], formed by the Ministry of Education in order to respond to the integration and exploitation of ICT in education [577]. It is noteworthy that all IT lessons take place in computer labs.

Students are given textbooks and educators have the right to use additional material. However, most courses are mainly lecture-based, following a traditional approach to teaching. Curricula information

[29] Hellenic Association of CS Teachers (PEKAP): http://pekap.gr/, last accessed January 2017
[30] Greek School Network: www.sch.gr, last accessed January 2017

and the provided books can be found at the official site[31] for ICT and CS in schools by the Greek Ministry of Education.

5.2 Method

The example used for the research is called "Computer uses in everyday life" which is a section from the CS book of the first grade of Greek Junior High Schools. This lesson had not yet been taught to the students of the school used in the present work. In the previous sections of the book, students had learned the key characteristics of computers and some basic software usage like word processor, spreadsheets and presentations. The chapter used illustrates the role of computers in different aspects of daily life. The purpose of this section was to enrich the students' prior knowledge and help them develop an integrated view of computer applications and the role of Information and Communication Technologies in the modern world. It was chosen because most students are familiar with this subject through their everyday life and could be easily engaged. Furthermore, both printed and internet sources have plenty of information about computers' usage. Finally, this chapter is ideal for collaborative activities, according to the textbook authors. The chapter consists of three sections that, based on the curriculum, are to be taught in two school hours. These sections are: a) Uses of computer in daily life, b) Computer's usage from distance, like tele-medicine, tele-education, e-commerce, e-governance and tele-work, and c) Risks from the use of computers in our lives.

A total of 57 students participated, 27 boys and 30 girls aged 12 to 13 years old, all of which were in the first grade (Greek Educational System) of Junior High School. They were from three different classes of the same school and for simplicity reasons for the rest of this paper they will be displayed as Group A, Group B and Group C. The selection was random and only based on the school's timetable. In addition, certain were kept constant between all three experimental conditions:

- ✓ students had not taught the specific course,
- ✓ teachers were present in all 3 conditions, and
- ✓ researchers had the same time (*2h*) with every group and under the same school conditions.

Table 9. Students by gender and learning group.

Experimental Condition		Gender of student: Male	Gender of student: Female	Total
Collaborative Learning	1st (Group A)	9	11	20
Peer Tutoring	2nd (Group B)	9	9	18
Traditional Teaching	3rd (Group C)	9	10	19
	Total	27	30	57

As presented below in detail (experimental conditions), students in group A worked in teams to create their own learning material on the given topic (not simply using their text book) and presented it any way they wished to their classmates (e.g. make a rap song, create a film, etc.). The best team from group A, would be the one to present the lesson they created to group B (the other class). Finally, group C was only exposed to a traditional lecture and only used information from their textbook.

[31] Digital Educational Content: http://dschool.edu.gr, last accessed January 2017

Details about each group are shown in Table 9 and are also presented in detail in the following sub-sections). The lesson was held in the computer lab which was set up in client–server architecture with 9 client PCs and one server computer used by the teachers, all with broadband internet connection. As for the additional technical equipment of the lab, there was also a projection system connected to the server computer.

The data gathered were both qualitative and quantitative. With the completion of each experimental condition students were given a handout consisting of the three following parts:

- ✓ Part I: Brief instructions for completion and acknowledgments by the research team.
- ✓ Part II: A test consisting of four short open ended questions from the school textbook for evaluating the learning results. The questions of the test were:
    1. Write briefly two examples of computer use in everyday life.
    2. What do we mean by the phrase "*The computer is a new way/mean of communication*"?
    3. Write three fields of computer usage that can be done by distance.
    4. What negative effects can computers have on our lives?
- ✓ Part III: A questionnaire consisting of five multiple choice questions about the research for capturing students' perceptions and opinions. The questionnaire is presented in the result section.

The test aimed to evaluate students' learning and was given to all three Groups (A, B, C). The two CS teachers of the school graded all the evaluation sheets of the students. To do this, teachers followed the instructions of the formal teacher's textbook, provided by the Ministry of Education. The questionnaire determined the degree of students' satisfaction of this alternative way of teaching and learning and was given only to Groups A and B where alternative teaching was used. Furthermore, the CS teachers of the school were interviewed about alternative teaching methods. The learning outcome, quantitative and qualitative results are analyzed and presented later in this paper.

In order to design the learning activities and evaluate the end result, the Bloom's taxonomy was taken into account [105] along with the revised version which includes 21st century skills [106]. The taxonomy has been widely recognized and used (both the initial and the revised form) in the design and evaluation of educational practices and educational technologies [578]. The different activities, described in detail below, correspond to the first two levels presented in the Bloom's taxonomy, namely "Remembering" and "Understanding". The taxonomy describes the different learning objectives in educational practices in order to provide a holistic and unified framework that would allow the common understanding and evaluation of educational approaches. The study was limited to the two levels due to time restrictions. Due to boundaries concerning the obligatory curriculum any activities had to be completed in limited school hours[32]. Only little deviations are allowed from the curriculum per academic year and per class. In Greece, when a researcher wishes to conduct a research or involve children in any activities she must obtain an approval from the Ministry of Education and the Pedagogic Institute[33]. Approvals take a significant amount of time due to bureaucratic issues. The researcher must have permission from: the director of the school, the school counselor and the teachers of the participating students. The teachers are responsible for the entire

[32] Circular of the Greek Ministry of Education: *F.3*/788/95795/G1/25-08-2011
[33] Greek Pedagogue Institute: http://www.pi-schools.gr/structure/departments/tetet/guidelines.php, last accessed February 2017

educational process and the safety of the children in the classroom. Thus the experimental conditions of the present research were concluded within three sessions and two school hours per session. For this reason, the study had to be completed within this strict time framework. Therefore, the method used focused only on the first two levels in Bloom's taxonomy. The next subsections present the three experimental conditions of our research. The first two were focused on active-learning methods and the third on traditional teaching (control group).

5.2.1 First Experimental Condition

The first condition consisted of two phases. In the first part the students of Group A worked together to create the course material in order to present it to their classmates. This phase examined the collaboration of the pupils and was conducted in two consecutive school hours. At first, the research team presented an outline of what students had to do. Students were asked to prepare a lesson of a specific section of their schoolbook and present it to their classmates as if they were teachers. For this reason, they would be divided into teams. The students collaborated at school, during that *2h* visit, but they could also work after school hours if they wished. The best team would be the one that would get the highest score from the students, following an anonymous voting process which is de-scribed in detail below. Most students were enthusiastic with the idea that they would present the lesson to their classmates. Extra motivation was also given by the fact that the best team of this class would present their work to another class of the school. Finally, the students were told that, at the end of this process, they would fill in a questionnaire about the process and would also answer a test sheet concerning their understanding of the lesson being taught.

The researchers briefly presented the three sections of the CS school book and made it clear to students that they should cover all the above in their teaching. In order to prepare their lecture, students were free to use any source they wanted such as textbooks, magazines, internet and others. Special emphasis was given to students' imagination as they were free to make their teaching any way they wanted using presentation software, lecturing, videos, documentaries, role-playing teaching or even by singing rap. Students were already familiar with word processing and presentation software and could use their knowledge to prepare a presentation for the course. Then the researchers and the teachers divided the students into four teams of five students each. The criteria for the division were based on the research on behavior of small work groups [579, 580]. The teams should have a proportionate number of females/males, should also have students of all grade levels (as their teachers pointed out) to create groups with all levels of abilities represented.

Each group used two computers for the remaining of the *2h* lesson of that day. Most of the students searched for information on the internet. Some teams started creating their presentation using PowerPoint software. The student's collaboration was flawless. The role of researchers, apart from answering a few procedural questions, was purely supportive and encouraging. As for the teachers, they had to intervene only in a few cases where students asked questions about the presentation software. Students showed such eagerness that when the school bell rang everybody stayed in continuing to work. Some teams split their work into subtasks so that all students would equally take part to things they could do. All groups cooperated perfectly and better students helped the most disadvantaged in parts where they had greater knowledge of the subject. As the CS teachers mentioned the weakest students had great motivation to participate op-posed to the traditional teaching. The collaboration process also seems to work well with students with special learning disabilities. In this class there were two students, a boy and a girl, diagnosed with specific mental learning difficulties (dyslexia and perceptual difficulties). Both were actively involved in the collaboration process. There is evidence from the literature, showing that collaboration is particularly effective for improving participation and retention of minority groups such as students with learning disabilities [581, 582]. Figure 41 and Figure 42 show students' teams working on their projects and the supportive role that their instructor had during the learning process.

Figure 41. Students working together.

The second phase of the first experimental condition was made a week after the first one. Students presented the lesson to their classmates and were evaluated by their peers in order to highlight the best work. Students worked entirely on their own without any essential help from their teachers. During the presentation of the teams, researchers and teachers were also able to ask questions as learners. At first most students were hesitant and shy so there was no discussion or any active participation. Thus, the research team recommended the students - instructors to ask questions during their presentation and seek participation of their peers. All the teams had prepared a presentation on PowerPoint[34], covering briefly all sections. Some teams additionally displayed YouTube videos about internet safety. The teaching of each team lasted about 15 min.

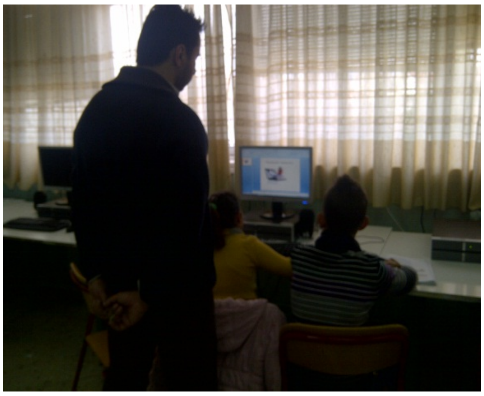


Figure 42. Teachers is just watching.

Some students were more capable in addressing questions to their peers and making them participate. It was also obvious also that some students were better prepared than others. It is noteworthy that the pupils of team D, unaided created a video using Windows Movie Maker software. The video presents different technologies, their evolution in time, social networks, possible dangers of technology and popular computer applications in daily lives. The students' video (without

[34] Students' works can be found online (in Greek) at www.ict.mysch.gr/?q=node/6

the narration which was done in class and in real time) is available online[35]. The students of team D used this video while they were talking to their fellow students explaining different aspects presented in the video.

Table 10. Teams' overall teaching scores.

	Team A	Team B	Team C	Team D
Overall Score	106	143	121	167

All the participants, i.e. students, teachers and researchers completed an anonymous evaluation form in order to assess teaching of the teams. Students were instructed to go beyond the surface of the presentations and try to assess the content as well as the appearance of the presentations. In addition, they were asked to evaluate the clarity of the information presented, the effort made by the students, how well they could deal with student questions, etc. The ranking score was from one to ten, minimum to excellent mark respectively. Students could not grade their own work and were told explicitly to be objective with their evaluations. A spreadsheet was used to display the results on board in an entertaining way. The spreadsheet updated and highlighted a field with the winning team, so at each entry the students cheered since the best team could change. Each teams' overall score is shown at Table 10. In general, it was observed that students enjoyed each other's teaching and assessed with high scores. Researchers and teachers were more contained, reflecting stricter standards. After the enjoyable scoring procedure, the winning team D presented once more its work, stress-free this time, in order to improve areas where deficiencies were observed.

5.2.2 Second Experimental Condition

In the second experimental condition the winning team of the previous phase per-formed their lecture to another class of the school (Group B). At first the researchers explained to Group B students the whole project. Students were already informed about the previous visits of the research team and expressed desire to participate in similar projects and complained about not being chosen for the first phase. Then, the teachers-students of Group A presented the lesson, better than ever, knowing exactly what to say and sharing roles among them. Researchers and teachers encouraged active participation and dialogue. Students of Group B made many questions regarding the presentation of their peers and seemed to be very competitive. There was a question where the teacher-students did not know how to answer, but a girl from the group calmly and with confidence, read from the textbook and responded. It was a proper reaction since a teacher could act in a similar way. In general, the students from Group A answered any questions made by their peers. Finally, they presented the video they had made, which excited the students and some asked to have it. With the completion of the second experimental condition, the students were given the same booklet as Group A, to assess Group's B learning, to evaluate the process and to compare results with the other student groups.

[35] Students' video about computers in everyday: http://video.sch.gr/asset/detail/r2aLOsZKOaGPDNZUKraYQC1l/MVfNhehQQ4CLESOm67swiSpu?language=en

5.2.3 Third Experimental Condition

This was the final phase of the experimental procedure and a member of the research team had to teach the same lesson to another class of the school (Group C, the control group). The students were required to use their textbooks certified by the Ministry of Education. These textbooks explain the lesson about '*Computers in our life*' using paradigms and colorful pictures, however, they do not have any student-centered activities. Students were passive participants during the lesson and they only listened and took notes as the teacher lectured. The instructor had seven years of experience as a CS teacher at Greek public schools. The lecture lasted one school hour. The fact that the teacher was unfamiliar to the students and students were aware that they participated in a study implied novelty [583] and Hawthorne [584] effects, which were indeed observed by the class teachers, since students showed both increased attention spans and effort levels. Due to the fact that such novelty effects were possibly also present in the previous experimental conditions (Group A and B), it explicitly decided that Group C should be taught by a person unfamiliar to them in order to maintain a novelty situation and try to balance out such effects through all the experimental groups.

The researcher-teacher used only the white-board and the school book during his teaching. He told students to read aloud specific sections through the book and then discuss all together around them. This mode is something customary in Greek schools from many teachers for children of this age. Then he made some questions to determine if students had understood the lesson. Again the teachers noted that most students were more careful than usual. He also gave some homework in order for them to have a considerable amount of unguided time and enable understanding and long-term retention of content. Finally, with the completion of this condition, the students were given the booklet that the previous students had completed. However, the booklet only consisted of the two first parts, not including questions assessing the novelty of the teaching method, since only traditional teaching was used in the case of group C (control group). In this case, group C was the control group, in a comparison between traditional teaching and collaboration and peer tutoring.

5.3 Results

Two statistical analyses were used for the results of this study. The first analysis utilized one-way ANOVA test, examining differences on the results of the three groups due to students' score to the tests. The second analysis used the nonparametric Mann–Whitney tests for comparison between perceptions of students due to the learning preferences with regards to satisfaction, difference, interest, learning and oddness among the active learning groups. A total of 57 students answered the survey and all answers were valid. The students of Group C did not answer the survey since they were not exposed to alternative teaching methods.

Table 11. Test Score - descriptive statistics.

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Min	Max
					Lower Bound	Upper Bound		
Group A	20	3,85	1,268	,284	3,26	4,44	1	5
Group B	18	4,61	,608	,143	4,31	4,91	3	5
Group C	19	2,68	1,455	,334	1,98	3,39	1	5
Total	57	3,70	1,401	,186	3,33	4,07	1	5

Table 12. Test of Homogeneity of Variances.

Levene Statistic	df1	df2	Sig.
5,622	2	54	,006

The results from the one-way ANOVA test show that statistically significant differences were found between the three experimental conditions (see Table 11 and Table 12). There was a significant effect on the score that the students achieved, F (2, 54) = 12.61, $p<0.001$ (see Table 13). For these data the variances are not relatively similar, as shown at Table 13, $p<0.006$. However, our sample is fairly small and this limits the Levene's test to detect differences between variances. Therefore, we can assume that variances are not homogenous and two more tests are reported, the Brown-Forsythe F (2, 42.69) = 12.92, $p<0.001$ and the Welch F-ratio, F (2, 31.99) = 14.85, $p<0.001$ (see Table 14).

Table 13. Test Score - ANOVA.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	34,997	2	17,498	12,610	,000
Within Groups	74,933	54	1,388		
Total	109,930	56			

Figure 43 shows a bar chart of the scoring data. The grading scale was similar to that of the schooling system. According to the rules for evaluation at the Greek Junior High School[36], the grading scale used is from 1 to 20. In addition, different grade ranges follow five specific descriptions. The degree is highlighted with the ratings: "*Bad*", "*Fair*", "*Good*", "*Very Good*" and "*Excellent*". The lower limit of each grading designation is defined as follows:

a) Bad → 01
b) Fair → 10
c) Good → 12 ½
d) Very Good → 15 ½
e) Excellent → 18 ½

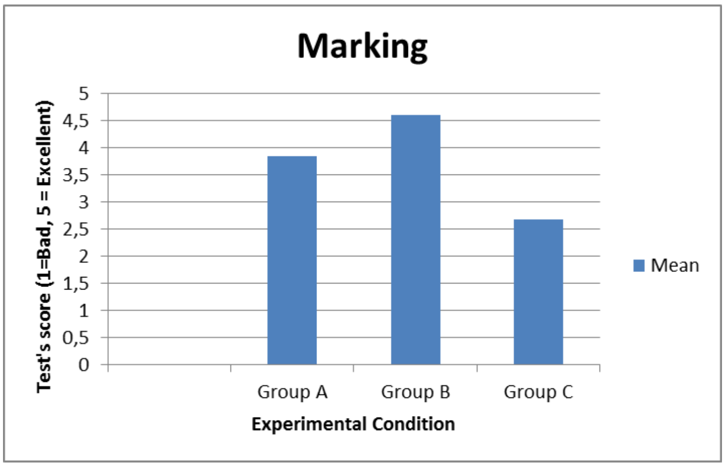


Figure 43. Means of Marking.

[36] Presidential Decree 465 of 11/15.5.81

Table 11 shows that students achieved better score at the peer teaching class (*mean 4.61*). Students of the collabo-ration class (*mean 3.85*) also achieved higher scores than students of the traditional learning environment (*mean 2.68*).

Table 14. Robust Tests of Equality of Means.

	Statistics	df1	df2	Sig.
Welch	14,851	2	31,992	,000
Brown-Forsythe	12,925	2	42,689	,000

a. Asymptotically F distributed.

After the completion of each experimental condition with active-learning, students were asked to report their satisfaction answering to the question "*How much did you like the course on computer usage in daily life?*" A simple three scale Likert scale ranging from 1 to 3 (1 being '*not at all*', to 3 being '*very much*') was chosen, for simplicity reasons and also following literature guidelines since the question was addressed to first grade students [585, 586]. We ran a Mann-Whitney's test to evaluate the difference in the responses of our 3-Likert scale question. We found a significant effect of Group U=144, Z=−1.98, $p<0.05$ (see Table 15 and Table 16). The mean ranks of Group A and Group B were *17.7* and *21.5*, respectively (Table 17). As shown in Figure 44 all the students of Group B were very satisfied with the lesson. Most students of Group A were very pleased with the lesson (*N=16*) and four (*N=4*) students were moderately pleased.

Table 15. Test Statistics[b] for Satisfaction.

	How much did you like this alternative way of teaching?
Mann-Whitney U	144,000
Wilcoxon W	354,000
Z	-1,979
Asymp. Sig. (2-tailed)	,048
Exact Sig. 2*(1-tailed Sig.)	,303[a]

a. Not corrected for ties.

b. Grouping Variable: Experimental Condition

Table 16. Descriptive Statistics for Satisfaction.

	N	Mean	Std. Deviation	Min	Max
How much did you like this alternative way of teaching?	38	2,89	,311	2	3
Experimental Condition	57	1,98	,834	1	3

Table 17. Mean Ranks for Satisfaction.

	Experimental Condition	N	Mean Rank	Sum of Ranks
How much did you like this alternative way of teaching?	Group A	20	17,70	354,00
	Group B	18	21,50	387,00
	Total	38		

Figure 44. Course Satisfaction.

Students were also asked to report the diversity of the lesson answering the question "*How different was this lesson compared to previous lessons?*" A simple three scale Likert scale ranging from 1 to 3 (1 being '*not at all*', to 3 being '*very much*') was again chosen. We ran a Mann-Whitney's test to evaluate the difference in the responses of our 3-Likert scale question. The two groups did not differ significantly U=171, Z= −1.98, p=0.761 (Table 18 and Table 19). The mean ranks of Group A and Group B were *19.05* and *20*, respectively (Table 20). As shown in Figure 45 nine students (*N=9*) of both Groups found the course very different. Eleven (*N=11*) students of Group A and nine (*N=9*) of Group B thought that the lesson was little different.

Table 18. Test Statistics[b] for Difference.

	How different did you think of this course in relation to previous?
Mann-Whitney U	171,000
Wilcoxon W	381,000
Z	-,304
Asymp. Sig. (2-tailed)	,761
Exact Sig. 2*(1-tailed Sig.)	,806[a]

a. Not corrected for ties.

b. Grouping Variable: Experimental Condition

Table 19. Descriptive Statistics for Difference.

	N	Mean	Std. Deviation	Min	Max
How different did you think of this course?	38	2,47	,506	2	3
Experimental Condition	57	1,98	,834	1	3

Table 20. Mean Ranks for Difference.

	Experimental Condition	N	Mean Rank	Sum of Ranks
How different did you think of this course?	Group A	20	19,05	381,00
	Group B	18	20,00	360,00
	Total	38		

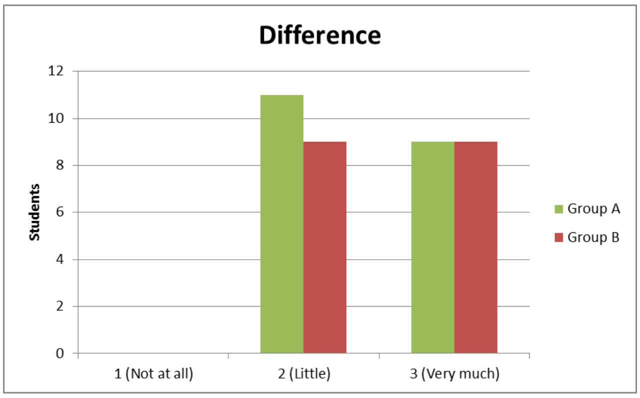


Figure 45. Course Diversity.

Students were also asked to report their interest in the lesson answering the question "*Would you like other IT lessons to be held in a similar way?*" They had to choose between two answers, yes or no. We also ran a Mann-Whitney's test to evaluate the interest in the responses of the students. The two groups did not differ significantly U= 163, Z=−0.935, p=0.35 (as shown at Table 21 and Table 22). The mean ranks of Group A and Group B were *18.65* and *20.44*, respectively (Table 23). As shown in Figure 46, seventeen (*N=17*) students of both Groups wanted to have other lessons taught in a similar way. On the other hand, only three (*N=3*) students of Group A and one (*N=1*) of Group B did not want similar teaching methods again.

Table 21. Test Statistics[b] for Interest.

	Would you like other CS lessons to be held in a similar way?
Mann-Whitney U	163,000
Wilcoxon W	373,000
Z	-,935
Asymp. Sig. (2-tailed)	,350
Exact Sig. 2*(1-tailed Sig.)	,633[a]

a. Not corrected for ties.

b. Grouping Variable: Experimental Condition

Table 22. Descriptive Statistics for Interest.

	N	Mean	Std. Deviation	Min	Max
Would you like other CS lessons to be held in a similar way?	38	1,89	,311	1	2
Experimental Condition	57	1,98	,834	1	3

Table 23. Mean Ranks for Interest.

	Experimental Condition	N	Mean Rank	Sum of Ranks
Would you like other CS lessons to be held in a similar way?	Group A	20	18,65	373,00
	Group B	18	20,44	368,00
	Total	38		

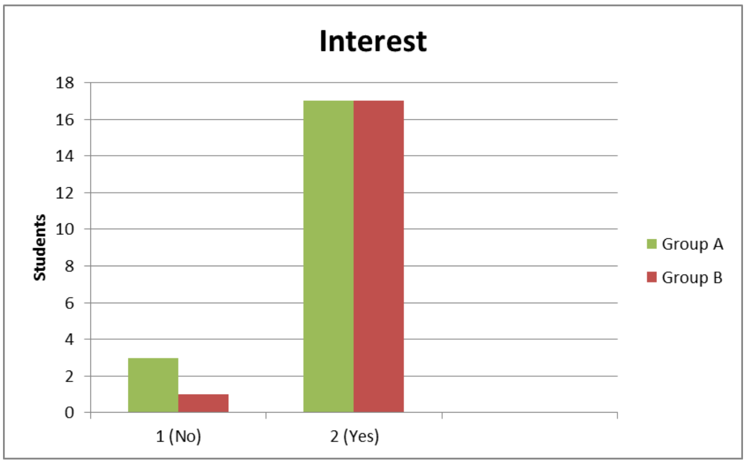


Figure 46. Course Interest.

Furthermore, students were also asked to report the perceived knowledge of the lesson answering the question "*How much do you believe that you learned compared to previous courses of CS?*" The simple three scale Likert scale ranging from 1 to 3 (1 being '*less*', to 3 being '*more*') was chosen. The Mann-Whitney's test was used to evaluate the learning of the students of Groups A and B. The two groups did not differ significantly U=146, Z=−1.26, p=0.206 (Table 24 and Table 25). The mean ranks of Group A and Group B were *21.20* and *17.61*, respectively (Table 26). As shown in Figure 47, sixteen (*N=16*) students of Group A and eleven (*N=11*) of Group B, believed that they gained more knowledge than traditional teaching. Four (*N=4*) students of Group A and seven (*N=7*) of Group B answered that their learning was about the same.

Table 24. Test Statistics[b] for Learning.

	How much do you believe that you learned compared with previous courses of CS?
Mann-Whitney U	146,000
Wilcoxon W	317,000
Z	-1,265
Asymp. Sig. (2-tailed)	,206
Exact Sig. 2*(1-tailed Sig.)	,331[a]

a. Not corrected for ties.

b. Grouping Variable: Experimental Condition

Table 25. Descriptive Statistics for Learning.

	N	Mean	Std. Deviation	Min	Max
How much do you believe that you learned compared with previous CS courses?	38	2,71	,460	2	3
Experimental Condition	57	1,98	,834	1	3

Table 26. Mean Ranks for Learning.

	Experimental Condition	N	Mean Rank	Sum of Ranks
How much do you believe that you learned compared with previous CS courses?	Group A	20	21,20	424,00
	Group B	18	17,61	317,00
	Total	38		

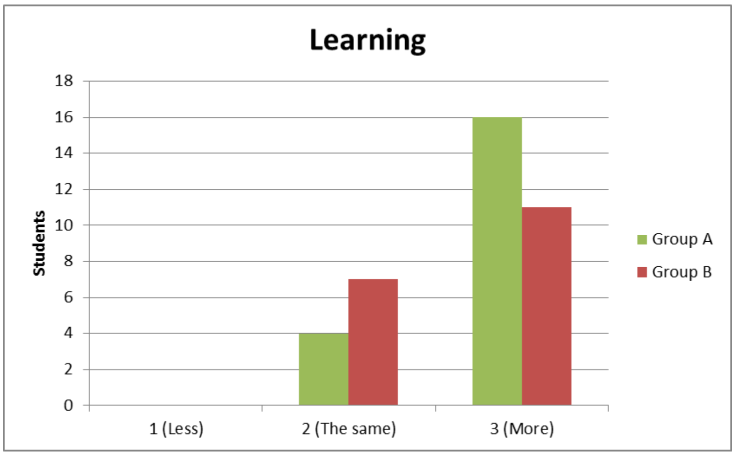


Figure 47. Perceived Knowledge.

Students were finally asked to report the oddness of the lesson answering the question "*Did you think that the way this lesson was made is odd?*" They had to choose between three answers, from a Likert scale (1 being '*little*', to 3 being '*very much*'). A Mann-Whitney's test was performed. Answers from groups A and B were compared. The two groups did not differ significantly U=170, Z=−0.327, p=0.743 (Table 27 and Table 28). The mean ranks of Group A and Group B were *20* and *18.94*, respectively (Table 29). As shown in Figure 48 three (*N=3*) students of Group A and two (*N=2*) of B found the lesson very odd. Eleven (*N=11*) students of A and ten (*N=10*) of Group B believed that it was a little odd. Finally, nine (*N=9*) students of both groups did not find it odd.

Table 27. Test Statistics[b] for Oddness.

	Did you think that the way that this lesson was made is odd?
Mann-Whitney U	170,000
Wilcoxon W	341,000
Z	-,327
Asymp. Sig. (2-tailed)	,743
Exact Sig. 2*(1-tailed Sig.)	,784[a]

a. Not corrected for ties.

b. Grouping Variable: Experimental Condition

Table 28. Descriptive Statistics for Oddness.

	N	Mean	Std. Deviation	Min	Max
Did you think that the way that this lesson was made is odd?	38	1,82	,652	1	3
Experimental Condition	57	1,98	,834	1	3

Table 29. Mean Ranks for Oddness.

	Experimental Condition	N	Mean Rank	Sum of Ranks
Did you think that the way that this lesson was made is odd?	Group A	20	20,00	400,00
	Group B	18	18,94	341,00
	Total	38		

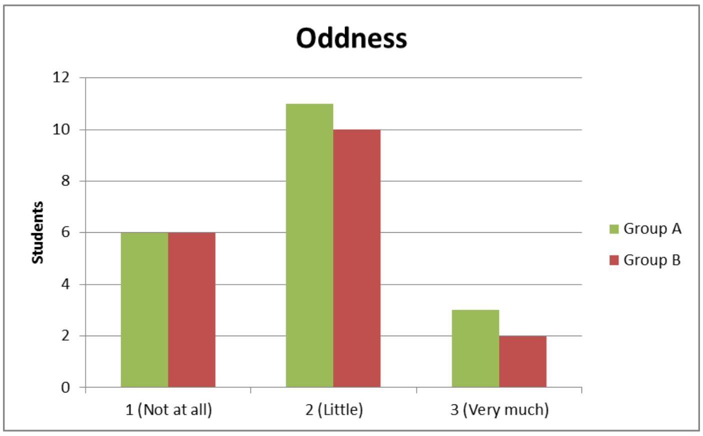


Figure 48. Course Oddness.

In addition to the quantitative results presented above, qualitative data were also gathered through teacher interviews. Teacher A is male aged thirty-five and teaches CS for seven years. He believed that students had greater commitment to the alternative lesson but he was not sure about the results at this point since it was the first time they did something like that. As for the students' work, he was pleased and he also noticed that the students collaborated willingly even outside school hours. Teacher A also believed that this teaching model could be applied more in CS education, because students comprehended some concepts better since they were actively involved. Yet, he thought that more time and experiments are needed to have a clear view of the usefulness of these methods. Nevertheless, he stated that the 1st experimental condition would have the best long-term results and that he would use active-learning methods in the future. Finally, he believed that many 21st century skills were enhanced, especially collaboration, communication and creativity.

Teacher B is female and aged forty-nine and has a 15-year experience in teaching CS. She believed that students learn better through traditional teaching, because they are accustomed to that way, although she found that students developed some new skills with alternative teaching methods. As for the quality of the students' works, she also found them very good and she was surprised by the fact that students worked even outside school hours. She also believed that this teaching model could apply only to specific sections of CS, like internet issues, software piracy, computer viruses etc. She pointed out that students liked to be in a leading role. Yet, she also believed that more experiments are needed to have a clearer view. She thought that the 3rd experimental condition can have the best long-term results but with very careful guidance from the teacher the 1st experimental condition could be also efficient. She also seemed willing to use active-learning methods in the future. Finally, she believed that some 21st century skills were revealed and developed, specifically collaboration and creativity.

Clearly there are differences between the two teachers' opinions. Teacher B prefers traditional teaching because she thinks it gives better learning results. Also she seemed to hesitate to include active-learning methods in her teaching, although see answered that she would use them under certain circumstances. On the other hand, teacher A was more excited by the results of alternative teaching and confident that he would use these methods in his teaching. A possible explanation could be attributed to the age difference of the teachers. The younger is more acceptant to challenges and changes than the senior. In addition, teacher B did not have a CS background in her studies (she holds a Physics degree) and that might be affecting her views and the way she handles CS classes. It is known from the literature that active teaching methods suits more confident teachers [587].

5.4 Discussion

In this chapter, different ways of teaching a theoretical CS lesson, to secondary school students using collaborative learning, peer tutoring techniques and traditional teaching were presented. The learning outcomes of the three settings and the student's perceptions were compared. Findings confirmed that active learning methods enhanced student interaction and preserved collective knowledge [124, 546, 557, 588].

The results of the present study provide support for such teaching methods in respect to learning efficiency. Students that participated in the alternative teaching sessions showed improved learning outcomes compared to those following traditional didactic instruction. CL and PT groups achieved highest scores at the test they were given, compared to the traditional teaching class. Collaboration process enabled individual students to tackle problems more efficiently once working in groups. It enabled them to share skills and knowledge which in turn strengthened their group. Although collaboration and participation was not assessed at the individual level (mainly due to the time restrictions explained above and also because many students collaborated outside school hours), it was observed that most students participated actively in the learning process and would like to repeat courses using similar alternative ways of teaching. It seemed that students wanted and needed to collaborate with their peers in ways that the traditional school system does not allow.

Apart from this, students also liked to be taught by their peers. The results of the study enhance the finding, that peer tutoring allows students to feel more comfortable and relaxed [565]. They did not seem to be intimidated as they might have been with a teacher, and may have been less hesitant to ask questions. Furthermore, students tutored by peers demonstrated an improved attitude in the classroom towards the subject matter. Current findings also confirmed that PT leads to greater comprehension of the lesson according to studies conducted [566].

Since Bloom's Taxonomy was used for the design of the learning activities (the first two levels only due to time restrictions), it was also used for the evaluation of the learning outcomes and particularly with respect to 21st century skills. The elements of the taxonomy together with their corresponding to the present study are presented in Table 30. In forthcoming works the design and evaluation of the learning activities will focus on higher levels of Bloom's taxonomy.

Table 30. Levels of Bloom's taxonomy used in this study.

Levels	Key Terms	Current Research Methodology	Current Research Evaluation
Remembering	retrieving relevant knowledge from long-term memory / **recall facts, define terms and concepts, locate information***	Students had to research for information at any source in order to complete their tasks during the learning evaluation. They had to define basic terms and concepts of "Computers' uses in daily life" and create their presentations by retrieving existing knowledge.	From the collected data during student evaluation, it was found that students were able to recall information effectively and answer the relevant questions. Students successfully managed to apply the gathered knowledge in the design of their group works and presentations. Those that participated in the active learning groups mastered the remembering level.
Understanding	constructing meaning from instructional messages, including oral, written, and graphical communications / **exemplifying, classifying, summarizing, inferring, comparing, and explaining**	At his level allowed students had to explain concepts and ideas they had to understand. Students had to link relative relationships between information to produce their final presentation.	In order to create the group works, students summarized and exemplified the information found. During both experimental groups (CL and PT) students also explained concepts to fellow students. They compared and evaluated each other works. The outcome of their projects showed that they were able to construct works of both educational and artistic value. The processes followed, confirmed students' understanding.

* Elements from the revised taxonomy [106] in Bold

This study suggested that students who participated in active-learning groups had better learning results. Students also believed that they learn more effectively when they participate to the learning process. Furthermore, students in the two experimental conditions showed increased levels of engagement, especially the students of the collaborative learning class. Students' satisfaction also increased their learning motivation as it was observed both by the researchers and the class teachers. It is also noteworthy, that students found the particular way of teaching a little peculiar, however, they still believed that it was interesting and worth repeating in the future.

In addition, positive perceptual changes were observed in regards to 21st century skills. In particular, in the two alternative teaching classes, the teachers observed positive student attitudes towards creativity, critical thinking, collaboration and communication. Saavedra and Opfer [551] also highlight the enhancement of essential 21st century skills among students who are exposed in alternative teaching.

Finally, differences were found between the teachers' perceptions about alternative teaching methods. It seems that some teachers were more open to adopt such methods than others. Felder and Brent [587], point out that those who are more confident may adopt more easily active teaching methods. In contrast, teachers preferring traditional teaching seem to believe that it leads to better learning results. Especially in traditional educational settings (like Greece), teachers hesitate to include alternative methods in their teaching [554]. Therefore, research on teachers' opinions and attitudes in active-learning can deepen further.

However, there are some limitations due to the specific section of CS that was taught and the number of students that participated. In this study, the participants were only from the first grade of a junior high school. They were taught a theoretical section of the Greek curricula about Computer uses in daily life. It is uncertain whether students of this age can teach others or collaborate in sections they do not have any previous knowledge or areas in computer science of a different nature like programming. As Michel et al. [589] conclude in their study, active learning is more appropriate once students already have a foundation in a particular subject matter, implying that alter-native teaching methods should be accompanied by a carefully designed scaffolding process.

Future studies should investigate perceptions of students from different grades of Secondary Education and conduct a more comprehensive survey of a larger student population. Finally, similar studies can attempt to evaluate the collaboration process and differentiate the individual and group work of the students. Collaborative activities that energetically engage the student and increase learning motivation should be further considered.

To sum up, further research based on empirical data is needed in the future, especially studies that investigate alternative teaching methods in traditional cultural settings, like the Greek educational system.

5.5 Conclusions

This chapter highlights the importance of introducing active-learning activities within the secondary education curriculum. In this study, a mixed technique based on alternative teaching methods of collaboration learning and peer tutoring was used. It seems that students conceived course concepts at a deeper level than the traditional teaching as well as being inspired to collaborate and be creative. However, being in a society that might not be entirely open to new approaches in teaching, the transition from completely traditional teaching practices to alternative ones should be done carefully and gradually. A balanced combination of both approaches (traditional and alternative) might be more effective in a changing educational system.

Apart from that, the effectiveness of alternative teaching methods is not yet deter-mined in other areas of computer science (e.g. programming) and the field remains largely open for novel research. However, it seems important to incorporate some collaboration activities in order to motivate students further. From the procedure described above, one may observe that the courses with alternative teaching methods were more attractive to the students. Through collaborative activities, students were able to better understand the lesson and, most importantly, to improve skills needed in the 21st century.

5.6 Elements in the ATMF

This section presents the elements of the ATMF that were affected by this research study (RS1). Figure 49 shows the teaching methods that were used. Peer Learning (Peer Teaching and Assessment) and Team Learning (Collaboration Learning) were implemented.

ATMs		
Element 2		Teaching
	Section 1	Peer Learning
	Section 2	Problem-Based Learning
	Section 3	Project-Based Learning
	Section 4	Studio-Based Learning
	Section 5	Inquiry-Based Learning
	Section 6	POGIL
	Section 7	Team Learning
	Section 8	Game-Based Learning
	Section 9	Educational Robotics
	Section 10	Non-Textual Programming
	Section 11	Contextualized Learning
	Section 12	CS Unplugged
	Section 13	Subgoal Learning
	Section 14	Programming Puzzles
	Section 15	Extreme Programming
	Section 16	Program Visualization
	Section 17	Competency-Based Learning
	Section 18	Social Networks Learning
	Section 19	Emerging Technologies

Figure 49. ATMs used in the RS1.

Figure 50 displays the elements of each component that were used in this study. From the first dimension, the "Infrastructure" element from component 1b is mentioned. Students worked in groups of four to five, due to limited numbers of computers. From the same dimension it is highlighted the component's 1c fifth element about "Time restrictions" that come from the educational system. Students that participated in the three experimental conditions of RS1, had limited hours available from school restrictions.

Dimension 1: Context		
Component 1a		Education Level
	Element 1	Formal Education / Type of School / Level
	Element 2	Informal Education
	Element 3	Non-formal Education
Component 1b		Environment (physical)
	Element 1	Infrastructure (availability)
	Element 2	Supporting Resources
Component 1c		Educational system
	Element 1	Policies
	Element 2	Compulsory Curriculum
	Element 3	CS standards
	Element 4	General public attitudes
	Element 5	Time restrictions

Dimension 3: Content		
Component 3a		Learning
	Element 1	Subject
	Element 2	Objectives
	Element 3	Short-term & Long-term Planning
	Element 4	Expectations for learning
	Element 5	Relations to other subjects / disciplines
Component 3b		Teaching
	Element 1	Lecture-based
	Element 2	ATMs (see Figure 35)

Dimension 2: Participants' Characteristics		
Component 2a		Learner related
	Element 1	Age
	Element 2	Gender
	Element 3	Cultural background
	Element 4	Socioeconomic status
	Element 5	ICT use
	Element 6	ICT experience
	Element 7	Attitudes
	Element 8	Personality traits
Component 2b		Instructor related
	Element 1	Age
	Element 2	Gender
	Element 3	Teaching experience
	Element 4	Academic characteristics
	Element 5	Skills - Abilities
	Element 6	Attitudes
	Element 7	Personality Traits

Dimension 4: Evaluation		
Component 4a		Learner related
	Element 1	Knowledge / Performance
	Element 2	Motivation
	Element 3	Skills
	Element 4	Metacognition
	Element 5	Beliefs
Component 4b		Instructor related
	Element 1	Assessment Criteria / Method
	Element 2	Monitoring of Learning (Learning Analytics)
	Element 3	Feedback to Learners
	Element 4	Long-term Evaluation
	Element 5	Teaching Evaluation

Figure 50. ATMF Elements of RS1.

From the second dimension it is highlighted the component's 2a seventh element about "Attitudes". Most students had a positive attitude towards alternative teaching and learning. From the same dimension and from the Component 2b, three elements are highlighted. These are the instructor related characteristics that seemed to affect the teachers' attitudes about alternative teaching methods.

Moreover, from the third dimension, the element of "Relations to other subjects" from component 3a is highlighted. The subject that was chosen for learning, is a broader topic. Furthermore, from the same dimension (3), both traditional lecture-based and alternative teaching were used (elements 1 and 2 from component 3b).

Finally, from the fourth dimension, four elements from component 4a were highlighted. About Element 1, the students' performance and gained knowledge were assessed. Their motivation and developed skills were also evaluated (elements 2 and 3). In addition, their beliefs (element 5) were investigated. From component 4b, the first element about assessment criteria is mentioned, since peer assessment was used as an evaluation method. It is clear that many other elements of the ATMF could be highlighted. Nevertheless, this section aims to present the factors that had a greater influence to the research study described in this chapter.

5.7 Summary

This research study compared learning results of alternative to traditional ways of teaching CS in Greek secondary education. Especially, PL and CL methods were used in secondary education students. The learning was assessed and students' attitudes towards alternative teaching were researched (quantitative data gathered). In addition, teachers' opinions were explored through means of qualitative data. The study aims to motivate adoption of student-centered learning techniques at CSE and encourage students to learn 21st century skills by doing. Therefore, it was used as an input during the design of the ATMF.

Chapter 6

By giving people the power to share, we're making the world more transparent.
Mark Zuckerberg

Research Study 2: Social Networks in Education

This study (RS2) reports on the use of the Facebook, as a supportive learning tool in Greek higher education[37]. Social networks (SN) seem to play a significant role in young adults' and university students' lives. Since students already use them in their daily life, why shouldn't formal higher education incorporate them? This chapter presents qualitative and quantitative data gathered through one academic term, after using Facebook as a teaching tool in Higher Education and investigates ways that SN can be used in teaching and learning. Issues of students' personalities, learning efficiency, motivation, cultural differences, gender differences and patterns of use are discussed.

The following sections describe the method used for data selection (both qualitative and quantitative data), the results found and the implication of those results for teaching and learning in higher education.

6.1 Introduction

Observing the first year students of the University of Peloponnese (average age 18 years) before the beginning of lectures, it seemed that most of them were using their Facebook accounts, if not solely, at least in parallel to other activities. Since these were computer science students, most of their teaching involves the use of computers in class. Talking to colleagues, it seems that students do not just use the computers during class only for their course work, but seem to be distracted with other activities, too. Use of Social networks (SN) was among the biggest issues and students seem to access those sites either from desktop computers of from their mobile phones. Since technology is there and students use it, how can we turn it from a class distraction to a teaching tool? This question leads to the development of the present work.

Moreover, as far as learning styles, cognitive styles and personality characteristics are concerned, it is known that any teaching approach (using technology or not) will benefit some students more than others [117, 590, 591]. Additionally, introvert students might engage more actively in mediated communication rather than direct and SN might provide the tools for that. Past research has shown clear indications towards this direction [592]. Using SN as an educational tool and in parallel to traditional teaching methods, the chances of benefiting students of different learning styles and personalities might increase. The present study also investigated issues of students' personality (cognitive style) in relation to the use of SN. Although cognitive style (to the knowledge of the authors) has not been used before to study Facebook's learning effectiveness, another tool has been used in the past to study possible effects on students' performance, namely communication style [438]. Cho et al. [438] showed how different communication styles affect learning performance. In particular, using SN in learning processes seems to favour students of particular communication

[37] Parts of this chapter have been published in: Antoniou A., Theodoropoulos A., Christopoulou K., Lepouras G. (2014). Facebook as teaching tool in Higher Education: A case Study. *International Journal of Advances in Social Science and Humanities,* 2(3), 43-56.

characteristics and the authors conclude that considering students' personal characteristics is crucial when SN are used in teaching and learning. For this reason, the present study included aspects of students' personalities when Facebook's efficiency as a learning tool was evaluated.

Furthermore, college students seem to prefer Facebook to other networks [446] and they also use it extensively. The faculty of the University of Westminster created and tested a specialized social network named "Connect" [593] to use by students and staff. Even when new innovative SN were tried, Facebook was still students' first preference. In another recent study, average time spent on Facebook by college students was around half an hour for the weekdays (active use), use was more common during evening hours and students reported that Facebook was a part of their lives, meaning that they would use it regardless of workload [435]. Particularly, Facebook was very popular with college students even more than any other social network [594].

However, SN and Facebook are not made for education. There are many other educational systems developed only for education with many desirable features, like forums, file sharing, video conferences, shared whiteboards, wikis, etc. [595]. Although, past studies have reported problems with the use of educational software, like learning style issues and software that does not match every preference, confusion from the use of certain features, like wikis, etc. [454], these systems are widely used.

Although SN and especially Facebook design evolved around purely communication concepts, they still seem to contain characteristics of effective learning platforms, as described in the literature [596]. For example, Facebook supports sharing, networking, community building, socialization, integration of newcomers, etc. In addition, past research shows that students are using SN more and more for educational purposes [433], among other purposes that remain primary, like socialization. Crook and Cluley [597] explored some tendencies of presentation and communication between Virtual Learning Environments and SN comparing 73 courses' platforms. The authors observed that the designers of typically learning platforms do not seem to take into account the significance of recreational internet services and the possible benefits of informal communication. Furthermore, students also report that the use of technology in formal education could increase learning motivation [598]. Using the Learning Ecology paradigm [599], learning does not necessarily occur only in formal educational setting and in school or college classrooms, but can be everywhere, anytime. Social networking can provide a link between formal and informal learning [600]. Therefore, any space, physical or virtual, that can provide learning opportunities is important. In addition, young people seem to be simultaneously involved in many different (learning) settings and learning spans contextual boundaries. In this light, SN can provide valuable opportunities for additional learning and as such they could be incorporated in formal education. According to different guidelines for necessary competencies for the 21st century learners, certain social network features like computer literacy, communication, effective information retrieval, are among the most important skills to be learnt [433, 600]. Moreover, previous studies have found that the use of SN can increase the quality of engagement and learning [601] and they provide opportunities for pedagogical mentoring [445]. Furthermore, students report positive perceptions for faculty use of Facebook and it seems that instructor engagement increases learning motivation through a relaxed learning atmosphere [602].

Yet, uses of Facebook in traditional educational settings are more challenging to adopt due to cultural factors [554]. Alike in Greek higher education, students and teachers expect certain behaviors and seem to prefer traditional situations. Therefore, the present work seeks to study different patterns of using Facebook among Greek university students. Together with possible cultural differences, students' personality traits, such as cognitive style are also considered. Students' opinions were recorded as well as instructor's notes and suggestions. In addition, the department of Informatics and Telecommunications, University of Peloponnese, uses a Virtual Learning Environment[38], widely used both by students and instructors which includes all the above characteristics. Since such specialized systems are primarily designed for educational purposes they should be superior to Facebook, but from their adoption in higher education, it has become clear that they are seldom used creatively [597]. Yet, students' behavior shows that Facebook has grown into an important learning platform for education [600]. Facebook provides the opportunities for immediate and quick communication between teachers and learners [600]. In addition, at Greek universities there is a common problem of low class attendance and using Facebook could work as a motivation for increased class turnouts, since it is a part of students' everyday life and they do not have to be actively involved with the course in order to get information. Based on these findings, it was decided to test the use of Facebook for educational purposes.

The present work is an observational study, wishing to reveal tendencies of social networks' use in higher education in a specific educational and cultural environment. The following sections describe the method used for data selection (both qualitative and quantitative data), the results found and the implication of those results for teaching and learning in higher education.

6.2 Method

Facebook was used as an additional teaching tool in the 1st year undergraduate module, Research Methodology in English for the students of the department of Informatics and Telecommunications, University of Peloponnese. There were 66 students registered in the course, 13 of which were females and 53 males. All students were between 18 and 19 years of age. Only first year students were used, since past research showed that especially first year students use SN to ease the transition from their homes to the new environment [603].

To use in the course, a new Facebook account was created, only for that particular purpose. Based on previous research findings, it seemed better to use a physical person identity, rather than create a special group, since many students report never to use groups and only to interact with individuals [435]. It was explained that all course necessary information would be posted also in the educational platform officially used by the university. Students that did not have a Facebook account were not expected to make one simply for the course. Students were free to choose whether they would participate or not, without any consequences on their course performance. From the 66 students registered, 14 students decided not to send a friend request, either because they were not using Facebook at all or because they did not want to participate, although they had a Facebook account. Students were also given a diary of use, in which they had to keep some notes and provide at least 5 logs describing a session of use. In each session they should record the time they spent, the medium (i.e. smartphone, pc, etc.), if they were alone or in a group, the aim (i.e. to communicate, to seek course information, etc.), their actions, possible problems they faced, possible solutions to these problems and other comments. At the end of the course (13 teaching weeks), students were given a questionnaire to provide both quantitative and qualitative data. Although past research showed that SN users are not always aware of their own behavior online and thus, it is not easy to collect reliable

[38] UoP eclass: http://eclass.uop.gr/, last accessed January 2017

data from questionnaires [433], in the present work, questionnaires were used in addition to qualitative methods for time efficiency and also because the questions did not ask students' perception on their patterns of use but only asked their opinion about the use of Facebook in class. Thus, the questions did not only ask for the estimates of use, but also about their opinion on different issues regarding SN, like feeling awkward communicating with the course instructor, etc. In addition, involving young adults might provide more reliable answers than having younger students. The questionnaire mainly consisted of open-ended questions.

Data were also collected from the instructor, by keeping a diary of use, together with student reactions to different posts and actions. Screenshots were kept of interesting interactions. From the instructor's diaries the following observations were made:

The instructor's posts were divided in 6 main categories:

- Procedural information about the course (i.e. marks now available on eclass)
- Reminders (i.e. please, bring your course books)
- Social postings (i.e. birthday wishes)
- General interest information (i.e. art installations, new paths in computer science)
- Evaluations and feedback aspects (i.e. questions like what do you expect from this course)
- Other notifications (i.e. information about university events, like free lectures, etc.)

Students' activities while interacting with the instructor were divided in 6 main categories:

i. Asking questions
ii. Asking for clarifications
iii. Replying to instructor's requests
iv. Replying to instructor's comments
v. Sending course work to instructor and other students
vi. Sending personal thoughts on social, political issues to the instructor

In order to study possible effects from the use of Facebook in students' learning performance and how that practice might be influenced by students' personalities, cognitive style was used. Cognitive style is a person's preference and habitual approach to the organization and representation of information [119]. Different researchers have described different aspects of cognitive style. The most common are field dependent-field independent [604, 605], impulsive-reflective [606], divergers-convergers [607], holist-serialists [608] and verbalizers-imagers [118]. Cognitive style is a research construct assisting the study of cognitive issues related to learning. It has a strong relation to the individual's personality and remains relatively constant over situation and time, or at least it is not that easily influenced by the different learning situations. A widely used assessment tool for cognitive style is the Myers-Briggs Type Indicator. The MBTI is based on Jung's theory of psychological types and it describes learners on four dimensions based on self-reported questionnaires. The dimensions are extraversion-introversion, sensing-intuition, thinking-feeling and judging-perceiving. The combination of the above dimensions provides 16 possible personality types, with different cognitive preferences and learning needs. MBTI has a strong validity and reliability, provided that the participants give honest answers. At the end of the course students were asked to complete a short version of the MBTI questionnaire.

Cognitive style was compared to students' final course marks. We used final course marks as an indication of students' learning. The issue is open to many interpretations and different methodological approaches, since other measures could be also used like skill building, added knowledge, etc. However, final marks have been successfully used in the past to study the learning effectiveness of different applications [438, 609-612].

6.3 Results

➢ **Popularity of Facebook**

From the questionnaire data, Facebook was indeed the most popular social network students used, since 51 out of 66 students reported using it (77.2%). Other networks used were Skype (22.7%), MSN (12.1%), Youtube (6%), from 4.5% networks like Twitter, Google+, Yahoo and ooVoo, and from 1.5% networks like Myspace, Gmail, X box live, Hi5, Livejasmine, Last.fm and Academia.edu. Moreover, for students that used multiple SN 87.2% reported that they used Facebook more than they used the others. Finally, only three students (4.5%) reported that they did not use SN. The reasons that they provided for their choice were: 1. "*SN are not necessary*", 2. "*I am not interested in SN*", 3. "*They trivialize social interactions*".

➢ **Privacy**

Although students sometimes have some privacy concerns, especially in regards to having a course instructor as their Facebook friend, it seems that their actions did not follow their verbal concerns. This finding is consistent in different social media studies with youth, across cultures and media, to the point that some researchers call it the privacy paradox [594, 613, 614]. As also found previously [435], students in this study provided free access to their personal lives and did not seem to restrict the instructor's access. In the questionnaire, one question asked whether they mind other students seeing details from their personal lives and only 19% of students either stated that they do mind (either all the times or sometimes). From the remaining 81%, only 21.5% mentioned that they do not mind since they never post personal details. Thus, almost 60% of students do not seem to mind for issues of privacy (Figure 51).

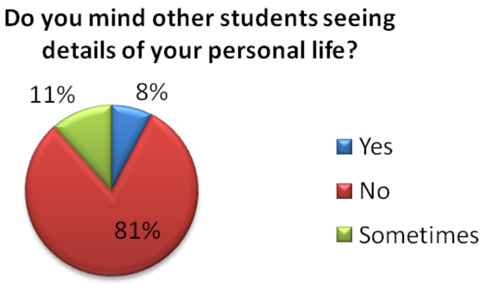


Figure 51. Students' responses about Privacy (I) on SNs.

Results seem to change a little when students are asked about lecturers seeing their personal lives. The following Figure 52 summarizes their answers.

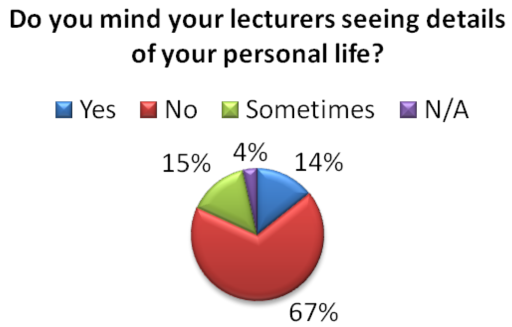


Figure 52. Students' responses about Privacy (II) on SNs.

In addition, in the question "Did the instructor's presence in Facebook alter your behavior online", 76.5% of students stated that they did not change their behavior. From the remaining 23.4% of students that did change their behavior, it is interesting to see some of their answers:

- ✓ "I was more polite"
- ✓ "I wrote comments in English, instead of Greek with Latin characters, but I expressed the same opinions"
- ✓ "…sometimes we were more discrete"
- ✓ "…now I pay more attention in blog articles and user comments".

Furthermore, students that decided not to send friend requests were asked at the end of the course for the reasons for this choice. Their answers can be classified in two themes. Most of those students mentioned that they had privacy concerns (46.1%) and some others that they are not frequent users of Facebook (15.3%). Privacy issues seem to have mainly affected their decision.

- **New friendships**

Previous research shows that college students using SN rarely make new friends online [603] and SN are used to support communication between people that know each other in the physical world. Other studies found that Facebook played an important role in the formation on new friendships for first year students and to make them feel settled at the university, since they use it both to keep in touch with their home friends but also to make new, at the new environment [446]. In the present study students reported that the module's Facebook page helped in making new friendships, since 48% of students answered positively in the relevant question.

- **Nature of posts**

Although students in previous studies reported that they wanted instructors only to post specific information [602], in this study a variety of posts were used, spanning from simple course instructions and procedural information to humoristic videos and birthday wishes (using the Facebook feature for birthday wishes). Moreover, 48.9% of students liked the humoristic videos posted by the instructor and 22.4 % likes the course related announcements. The following Figure 53 shows students' preferences on instructor's posts.

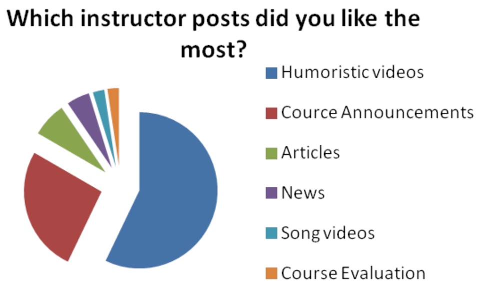


Figure 53. Students' responses about nature of posts on SNs

➢ Gender differences

Gender differences in the use of SN have been found before [615]. The instructor also observed significant differences in the use of Facebook between girls and boys. Not only the post topics were different, with boys talking about sports, motorcycles, etc. and girls talking more about feelings, relationships, fashion, etc. but also more girls than boys posted personal information. In particular, it was mostly girls that informed their wall with details of their personal lives or used offensive phrases, whereas boys seem to be more careful. In addition, in the questionnaire students were asked to identify female and male inappropriate behaviors on Facebook.

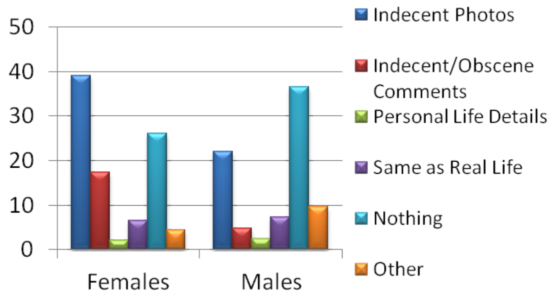


Figure 54. Gender differences in the use of SNs.

Their answers show that not only different genders have different behavior on SN but they are also expected to have different behavior. Figure 54 displays their answers.

➢ Academic and real life gap

One of the main hypotheses of the present work was that Facebook could bridge the communication gap between students (especially first year students) and course instructors, a common problem also acknowledged in the literature [595]. In the questionnaire, there were three relevant questions. In one question, 59.5% of students stated that Facebook helped them feel closer to the instructor. Two more questions were used, one asking them if the course's Facebook page made them feel closer to the University and the Department and the second if Facebook could be used to bridge private and university life. In the first question, the majority of students reported that the use of Facebook did help them feel closer to the University. The following Figure 55 summarizes their answers.

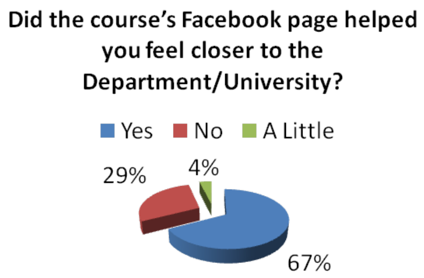


Figure 55. Academic and real life gap (I) with Facebook.

In the second question asking if Facebook could bridge the private and university life, 61.2% of students gave a positive answer. The following Figure 56 summarizes their answers.

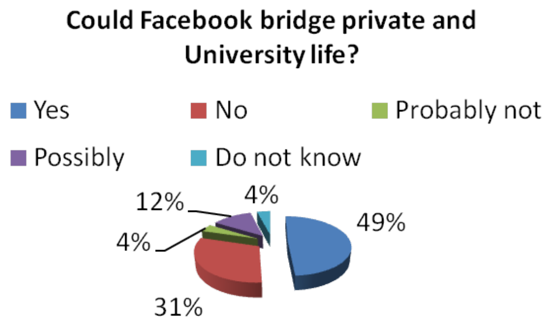


Figure 56. Academic and real life gap (II) with Facebook.

➢ **Motivation**

Another main hypothesis regarded student motivation. This was split into two questions: the first question asked whether the use of Facebook increase their lecture attendance motivation, since this is a serious issue at the department and often classes show low turnout rates. In this question, about half the students (42.8%) stated that their class attendance motivation increased due to Facebook. However, the instructor observed a significant increase in class attendance (that of course could be due to other factors as well), but the class average absence rate was about 1.6 absences per student for a 13 lectures course. For this particular issue, two student comments are notable:

➢ "*SN must be used in classes not because they only help in the better functionality of the lesson, but because they work as class attendance motivators.*"

➢ "*One general comment about the module: in my opinion if this course was done differently (for example traditional lectures) the class attendance would be minimum.*"

Both comments show that students also recognize the motivating power of SN in learning processes.

The second question on this issue, asked directly on students' perception of Facebook increasing their learning motivation overall. Only 38% of students provided a positive answer to this question.

➢ **Course evaluation**

Both in class and in Facebook, the course instructor asked students to provide their midterm evaluation, together with their suggestions for the improvement of the course as a whole. All the students that sent their evaluations did it through Facebook. In a relevant question at the end of the term in the questionnaire, students reported that Facebook could help course evaluation processes. The following Figure 57 demonstrates their answers.

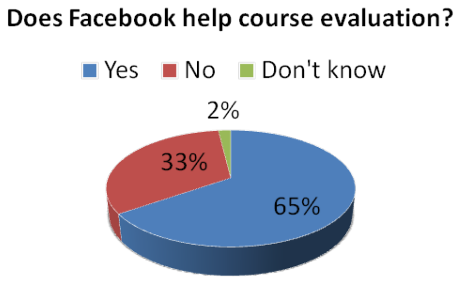


Figure 57. Course evaluation with Facebook.

➢ **File/Information exchange**

Facebook was used frequently for the exchange of course related files and information, since 56 % reported that they used it often for these purposes, 30% rarely and only 14% never used it in such ways.

➢ **Communication preferences**

Effective group communication relies both on public and private conversations [616] and Facebook is good at providing both. In fact, students used both in contacting the course instructor. Students preferred using Facebook for their communication with the course instructor, more than mail and face-to-face interaction. Unlike previous findings that indicate that the use of communication technologies traditionally used in colleges is similar to the use of Facebook for academic communication purposes [445], in this study students by far preferred to use Facebook and the messaging option to emails. In addition, in the questionnaire answers students reported that this kind of communication is beneficial especially for introvert students, or as they put it "shy" students. This is a similar finding to previous research [617].

➢ **Student diaries**

Although students were instructed to complete diaries of use, only eight students returned their diaries and only four were fully completed. Thus, data from these diaries are very restricting and cannot be used. The limited completion of diaries was probably because students had to complete them at home while using Facebook. A different methodology will be used in a future study to collect rich qualitative data (i.e. possibly photo-diaries during class, etc.).

➢ **Course structure and cognitive style**

In an attempt to study the effect of the use of alternative teaching methods through SN in class, students' personality characteristics were compared to their final course marks. Knowing that different teaching methods might positively affect students of certain personalities, students' cognitive styles were compared to their final course marks. Although, using course marks for this comparison might not be a good indicator of actual learning, in a quantitative methodology (used on this case) it is difficult to collect different data. Therefore, a course mark is treated here as an indicator not as an absolute learning factor. Data from 49 students were used to perform the statistical analysis. Since the number was small, the cognitive styles were deconstructed to their initial dimensions. MBTI can provide 16 possible personality types, reflecting four dimensions. In the present analysis, we used the four dimensions giving different values for the two options. Similarly, marks were given in a scale of four categories (fail, pass, second, first). Since all data were categorical, a Pearson's Chi Square test was performed. When marks were compared to the Extraversion-Introversion dimension the value of $x^2(3) = 1.549$, $p=.05$. For the Sensing-Intuition dimension and marks, $x^2(6) = 4.207$, $p=.05$. For the Thinking-Feeling dimension and marks, $x^2(3) = 7.283$, $p=.05$. For the Judging-Perceiving dimension and marks, $x^2(3) = 1.014$, $p=.05$. Therefore, no statistical significance was found in any of the tests performed.

6.4 Discussion and Conclusions

- **Popularity of Facebook**

Facebook at present seems indeed the most popular social network used by university students and particularly in the present case study. However, SN follow fashion. Facebook is fashionable and popular for now, but it will not be forever. Instructors that decide to use SN should often verify the actual use and popularity of the tool they are using. Since, one of the main arguments for the use of SN is their popularity among students, it is desirable to maintain this advantage.

- **Privacy**

The privacy paradox was observed in this case study as well as previous studies. Young students do not seem to mind other, including instructors to know details of their private lives. From our observations is also seemed that male students were more protective to their personal lives than female students were. Female students were more likely to upload dating photographs, discuss openly about their romantic involvements and so on. It seems that the use of SN opens a new study field in anthropology and results from future anthropological research could further inform the educational potentials of such systems. The following Figure 58 shows such examples.

Figure 58. Privacy examples on SNs.

- **New friendships**

Previous research has shown conflicting evidence in relation to the issue of making new friends on SN. There might be cultural differences concerning this matter and it is worth further future study. According to the participants of this study, the course's Facebook account did help them make new friends. The following Figure 59 from the instructor's diary of use provides an example, showing two students of the course connecting on Facebook.

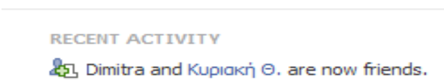


Figure 59. New friendships on Facebook.

➢ Nature of posts

The present study attempted to bridge formal and informal learning, by posts of varying themes, since there is limited research in informal learning and the use of modern technology [432]. As explained above, previous studies showed that students wanted specific and directly course related information to be posted by instructors [602]. Various posts were used in this study spanning from formal learning material to birthday wishes and humoristic videos. Students reported liking the different natures of the posts, mostly favoring humoristic videos. The following Figure 60 shows students' reaction to an art video.

Figure 60. Reaction to art video on course's Facebook page.

Figure 61 shows a student's response to Birthday wishes.

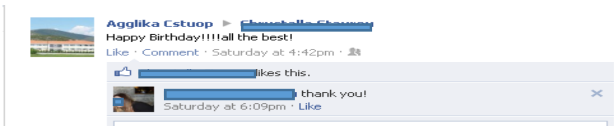


Figure 61. Reaction to birthday wishes on course's Facebook page.

In addition, the following Figure 62 shows reaction to humoristic photos.

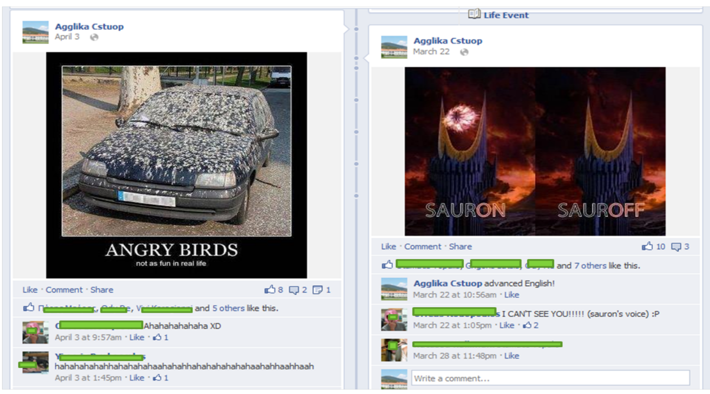


Figure 62. Reaction to humoristic photos on course's Facebook page.

The student data showed that not only students enjoyed the different posts, but they also viewed them as motivators for further engagement: "*Yes, for example, student responses to the module's status, attracted our interests and provided motivation for further engagement*". Even some of the humoristic posts engaged students in a richer dialogue, involving political views and historical facts. For example, due to an upcoming football match between Greece and Germany the instructor posted a humoristic video. However, this triggered a serious conversation about human morals and ethical values, starting from a historical event from World War II and Ukraine's national team defeating the Nazis and therefore losing their lives. Facebook in this case allowed a very interesting conversation to develop that due to time restrictions would be almost impossible to happen in a physical lecture theater. In a way, there was a clearer connection between formal and informal learning in this case, that effectively used an informal medium (such as Facebook) to engage people usually interacting in a formal learning setting (students and instructor) in an informal learning conversation (human moral values). The following Figure 63 displays the students' initial responses to the humoristic video and the next story that followed.

Figure 63. Reaction to humoristic video on course's Facebook page.

Knowing that it is more difficult to maintain relationships online and it takes significantly more effort [618], means that instructors using social media need to be very active, since they have weak ties with the students. The use of social media in class, changes the very nature of learning and teaching, implying that teachers need to practice new ways of approaching students, allocating tasks, distributing information, etc. In addition, using SN implies that students have access to a wide range of resources. In this course, the instructor tried to use different sources and material, not simply course instructions or only course directly related material. However, using SN the instructor runs the risk of harming her credibility with the students but if the networks are used appropriately the instructor can signify that she understands student culture [602]. Finding the balance between the two is not always straight forward and the instructor needs to remain aware of the potential risks and also the benefits [446]. For example, in this particular study, a way to tackle the problem and to adopt a less intrusive profile, the instructor chose not to comment on students' posts, but only to comment under her posts and student reactions to those. The students seem to acknowledge the effort made by the instructor to follow a similar style of use of Facebook as their own. One female student mentioned: "...you tried to follow the way we use Facebook". The nature of posts used in Facebook for educational purposes is another big area opening that requires future study and since the present study found different results compared to previous works, cultural differences might be interfering.

- **Gender differences**

Similarly, gender differences also require further study. It seems that there are strong stereotypes and expectations about female and male use of SN. It seems more unacceptable for girls to post indecent photos and obscene comments than boys do. Men are more allowed to express themselves freely than women do. On the other hand, different genders seem to be interested in different topics and women, at least in this study, were more likely to post details of their personal lives.

- **Academic and real life gap**

It is also interesting to note some of the students' further comments in the question about Facebook bridging University and private life, although most students agree that it can be used to bring the University closer to their everyday lives and bridge the gap. One student mentioned "*Facebook can be a bridge but not a very solid one*". This is a very good point that could be further extended to learning. SN can be social, learning, communication and other tools but they need to be used in specific frameworks. On their own, they cannot provide very solid basis for substantial interactions. Keeping that in mind, using SN in learning requires a very different and active role from the instructors. Learning does not stop in class but it follows us in our homes and private lives, where students can contact us any time and require diverse information and interaction.

- **Motivation**

Concerning learning motivation, although 58% of students provided a negative answer, it is interesting to look at some qualitative data from their comments:

- ✓ "*Yes (Facebook increased learning motivation), because it was easier to communicate with my fellow students – exchange of information*"
- ✓ "*Yes, because it motivated me to look for new knowledge*"
- ✓ "*Yes, because when I saw something (in Facebook), I always googled it immediately*"
- ✓ "*… to a certain degree, it provided daily contact with the module*"
- ✓ "*Yes, because you tried to use Facebook the same way we do"*
- ✓ "*Yes, because the course was more pleasant and active*"
- ✓ "*Yes, because there was communication through Facebook*"
- ✓ "*Yes, because we could follow the progress of other students, too*"
- ✓ "*It was helpful, because even by chance I was involved with the module more than twice per week*"

It seems that Facebook has some learning motivating powers, although students do not always recognize them. Student comments reveal that Facebook can address a number of learning issues, like comparison of one's work to the rest of the class, easy communication with instructors and fellow students, connection between formal and informal learning, constant engagement with the course, etc. However, the methodology used in the present work did not allow further study of this issue, since it could only rely on students' perception of learning motivation and indirect measures, like absence rates. To study the matter deeper, we would probably need to repeat the course next year (keeping the learning material consistent) with new students but the same instructor (eliminating instructor personality differences) without using Facebook and compare class average marks.

Moreover, Madge et al. [446] claim that using Facebook not only motivates students but also instructors. From our observations throughout the duration of the course, this seemed to be very true. As far as the instructor is concerned, the motivation level of engagement with the course increased. Facebook provided an insight into students' lives, problems, concerns and interests. It thus provided an excellent source of material for the instructor, since it could be used in class for teaching purposes. For example, after observing in Facebook that students communicated by posting music videos, music was used in the class for different activities (i.e. translate your favorite English song, during English lessons). Students were very enthusiastic with activities that incorporated their interests.

- **Course evaluation**

Facebook allowed direct and constant course evaluation, either by students' answering direct questions or by the instructor observing their reactions on Facebook during class. In addition, the evaluation processes go both ways, since students can also be quickly informed about their performance, as well. The informal nature of Facebook allowed imaginative ways to provide evaluation to the instructor and to the students. For example, the students responded with song videos to the course midterm evaluation request and the instructor provided marks for best course work in the form of Oscars. Using the Oscars metaphor, two needs were covered to praise good course work but also to provide public recognition. Figure 64 shows the number of students' reactions, together with posts of various song videos.

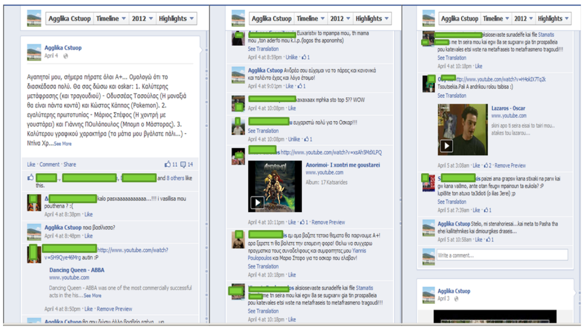


Figure 64. Course work evaluation in the form of Oscars.

- **File/information exchange**

Students were using Facebook to exchange information and course related files with their classmates. They reported seeing it as a good tool for that purpose. In addition, using Facebook after teaching and office hours gave students more freedom in their communication with the instructor. From our teaching experience, students rarely send uncompleted course work to be checked by the instructor using other means. In previous years when we were not using Facebook this was the case. This seemed to change when Facebook was introduced. Students were sending more questions (especially when they could see the instructor online by using chat) and uncompleted course work. The following Figure 65 shows a student asking for feedback while working on his assignment.

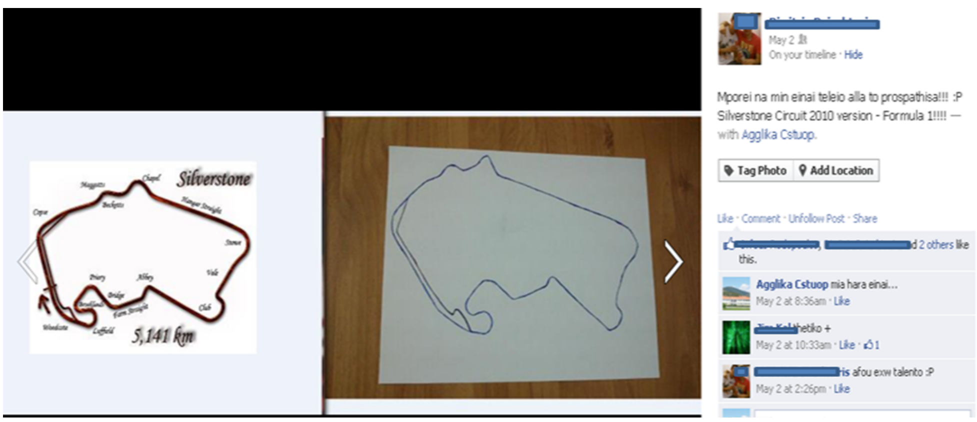


Figure 65. Evaluation on work in progress.

➢ Course structure and cognitive style

No statistical significance was found when different personality dimensions were compared to the students' final marks. These results might be due to various reasons. One reason might be that the final course marks used might not reflect the actual learning. In any case, the current results show that the specific course structure and teaching style do not seem to favor particular cognitive styles. Although these findings have not been crosschecked with classes that students did not use Facebook or other alternative teaching methods, the present results could imply that using Facebook as an additional teaching tool to the traditional lectures might be an inclusive factor to students' different cognitive styles. Knowing that using a particular teaching method favors certain students more than others depending on their personalities, using alternative teaching tools, such as SN might increase the number of cognitive styles favored. The present results are by no means conclusive but they rather indicate that further research is required in this direction.

➢ Other points

Using Facebook to interact with students as a physical person and not as a group, although increased the level of interaction between the instructor and the students, also runs the risk of the expression of inappropriate behaviour or the level of perceived closeness to the instructor. Having used a group for that purpose would have been a safer option, since students are aware that everything they post is visible to the rest of the group. However, in this particular study and for reasons mentioned above (students rarely interact with groups) the physical person approach was applied.

SN can be used for university marketing campaigns, since they are highly effective [617] and also help students to fulfil their necessity of belonging to a community out of the class [597]. In the present study, for example, Facebook was used to advertise free lectures (Figure 66), by using different options (i.e. posts and events planner option).

Figure 66. Announcing events on course's Facebook page.

Students spend more time reading than directly acting [435, 619]. In our study, students reported that they watched most of the videos posted by the instructor, although they rarely wrote any comments. In particular, 37.5% of students stated that they watch the different videos posted by the instructor but they did not post any comments. Therefore, the instructors that decide to use Facebook as a teaching tool, should not be discouraged if students do not seem to respond to posts.

Moreover, identity formation is an essential part of youth and young adults seem to use SN towards that end [435]. For example, students customize backgrounds, post favorite music videos, personal thoughts, etc. Previous studies have shown the importance of social media in expressing personal identities and the finding is robust across media and geographical areas [433]. In particular, music seems to play a very important role in the expression of identities and different studies support this argument [433, 435]. This feature of social media that allows students' identity formation could be used for educational purposes (i.e. involve music in different assignments).

Students were also asked to describe their feelings from the experience. Many students mentioned that it was a positive experience, that they felt comfortable, or that they handled the instructors'

Facebook presence as any other friend. No student provided any negative feedback in regards to their feelings, although a couple mentioned that they felt a little strange at the beginning but not in a negative way. It was interesting to read that some students mentioned that they felt special having a course instructor as their Facebook friend and as one of them put it: "*I felt special, because this does not happen in many Universities*". Another student said that he felt proud and another one that "*I felt that this broke the established old fashioned rules for the relationship between students and instructors*". Another similar point is: "*…through their actions (instructors'), you see them as friends and not as professors/dictators*". Two more students started that they felt closer to the instructor and motivated to learn. Finally, a girl mentioned that although she did not feel anything different, this process made her realize that the instructor wanted to approach the students. All the above comments are very interesting and show the need for affective learning in Higher Education. Although this need in recognized by educators, it is often neglected [620, 621]. Therefore, a future study could focus on the possible connection between SN and affective learning.

In a final question, students were asked to make suggestions for the use of SN in their department. Only one student thought that the use of SN can have distracting powers and should not be used in formal learning. All other students were positive and here are some of the most interesting answers:

- ✓ "*…they provide (i.e. chat option) an easier and direct contact between students and staff compared to email*".
- ✓ "*I would like more students to participate actively*".
- ✓ "*It could be improved with more humoristic videos, photos from class and more announcements in general*".
- ✓ "*…you could record the lectures and upload lecture videos on Facebook*". A few students suggested this particular idea.
- ✓ "*…instructors become more active and they motivate us to engage more*". There were similar comments from a few students.
- ✓ "*…there are specialized educational networks to use in the department and they could be linked to Facebook*".
- ✓ "*…the (course) announcements saved as from checking our webmail. In addition, from the songs and videos the students form a different opinion for the instructor*".
- ✓ "*I wished more instructors used Facebook…*".
- ✓ "*They (SN) improve the cooperation between students and staff and thus improve the quality of work*".
- ✓ "*They should have a discussions forum only visible to students*".
- ✓ "*It was a good way to compare opinions and to exchange files and reports*"

Finally, for the development of this work, students were asked to participate in the data analysis and writing of the report through Facebook (asked for volunteers). The third author of the present work is one of the first year students that responded to the request and most of our cooperation and paper preparation went through Facebook (personal messages and chat), i.e. exchanging data files, versions of the draft, messages about procedural issues, etc. Being in the research world for some years, this is the first time we use SN for not only data collection, but also for researchers' communication and file exchange. Usually, this work was done with other tools, like Skype, mail and Dropbox. Working with a young student, brought Facebook in our research tools reservoir and the experience was positive. Therefore, SN might need to find their way into Higher Education by not simply assisting teaching but also research.

As a final thought, a student's comment is quoted: "*(Facebook use is formal education) is at an early stage*", implying that e-maturity ("*the extent of provision, management and use of technology to support learning across the* *curriculum*" [622]) is a crucial aspect. Teaching stereotypes and traditional expectations seem to apply for both students and staff. In addition, not all students seem ready to actively adopt SN for their formal learning [623]. However, we need new pedagogical approaches that enable learners to change identities with technology being only a medium. Until e-maturity regarding the use of SN is reached, we can only keep on studying the field and experiment with the use of different networks.

6.5 Elements in the ATMF

This section presents the elements of the ATMF that were affected by this research study (RS2). Figure 67 shows the method that was used additionally to the traditional teaching.

ATMs		
Element 2		**Teaching**
	Section 1	Peer Learning
	Section 2	Problem-Based Learning
	Section 3	Project-Based Learning
	Section 4	Studio-Based Learning
	Section 5	Inquiry-Based Learning
	Section 6	POGIL
	Section 7	Team Learning
	Section 8	Game-Based Learning
	Section 9	Educational Robotics
	Section 10	Non-Textual Programming
	Section 11	Contextualized Learning
	Section 12	CS Unplugged
	Section 13	Subgoal Learning
	Section 14	Programming Puzzles
	Section 15	Extreme Programming
	Section 16	Program Visualization
	Section 17	Competency-Based Learning
	Section 18	Social Networks Learning
	Section 19	Emerging Technologies

Figure 67. ATMs used in the RS2.

Figure 70 displays the elements of each component that were used in this study.

Dimension 1: Context		
Component 1a		**Education Level**
	Element 1	Formal Education / Type of School / Level
	Element 2	Informal Education
	Element 3	Non-formal Education
Component 1b		**Environment (physical)**
	Element 1	Infrastructure (availability)
	Element 2	Supporting Resources
Component 1c		**Educational system**
	Element 1	Policies
	Element 2	Compulsory Curriculum
	Element 3	CS standards
	Element 4	General public attitudes
	Element 5	Time restrictions

Dimension 3: Content		
Component 3a		**Learning**
	Element 1	Subject
	Element 2	Objectives
	Element 3	Short-term & Long-term Planning
	Element 4	Expectations for learning
	Element 5	Relations to other subjects / disciplines
Component 3b		**Teaching**
	Element 1	Lecture-based
	Element 2	ATMs (see Figure 33)

Dimension 2: Participants' Characteristics		
Component 2a		**Learner related**
	Element 1	Age
	Element 2	Gender
	Element 3	Cultural background
	Element 4	Socioeconomic status
	Element 5	ICT use
	Element 6	ICT experience
	Element 7	Attitudes
	Element 8	Personality traits
Component 2b		**Instructor related**
	Element 1	Age
	Element 2	Gender
	Element 3	Teaching experience
	Element 4	Academic characteristics
	Element 5	Skills - Abilities
	Element 6	Attitudes
	Element 7	Personality Traits

Dimension 4: Evaluation		
Component 4a		**Learner related**
	Element 1	Knowledge / Performance
	Element 2	Motivation
	Element 3	Skills
	Element 4	Metacognition
	Element 5	Beliefs
Component 4b		**Instructor related**
	Element 1	Assessment Criteria / Method
	Element 2	Monitoring of Learning (Learning Analytics)
	Element 3	Feedback to Learners
	Element 4	Long-term Evaluation
	Element 5	Teaching Evaluation

Figure 68. ATMF Elements of RS2.

The use of Facebook, involved learning both in formal and informal settings (elements 1 and 2, component 1a, dimension 1). However, Facebook might be a distraction tool from the learning process and thus many are opposed to its usage in education (element 1e, component 2a, dimension 1).

From the second dimension are highlighted the component's 2a elements about "Gender" (2) and "Cultural background" (3). RS2 found some gender differences from the use of Facebook and moreover the traditional Greek educational system stays as an important factor about such alternatives. From the same dimension, about instructor related elements, attitudes about alternative teaching methods are mentioned.

Finally, from the fourth dimension, two elements from component 4a were highlighted. Element 2, about the students' motivation and element 5 about students' beliefs. From component 4b, the third element about feedback to learners is mentioned, since Facebook was also used for that purpose.

6.6 Summary

This chapter presented an observational study, that revealed tendencies of social networks' use in higher education in a specific educational and cultural environment. Furthermore, Facebook was used as a teaching tool in higher education and ways that SN can be used in teaching and learning were investigated. The study reported that students were highly motivated to participate in the lesson through the Facebook page although there seem to exist some personalities, cultural and gender differences about the usage.

Chapter 7

Game Based Learning is the future of education.
Bill Gates

Research Study 3: Game Based Learning

This chapter [39] is a case study on the use of educational digital games to elementary pupils and deals with games that promote algorithmic thinking as well as the basic principles of programming. The study (RS3) involved 94 young pupils who worked in pairs (Pair Programming) trying to solve puzzle games from code.org website during their visit to the University of Peloponnese in the context of the European Code Week. Factors such as satisfaction, evaluation of perceived knowledge and willingness to use games in future in the learning process were examined. Furthermore, it's discussed whether a short activity like this can result to long term motivation of pupils. The chapter concludes that: (a) the pupils enjoyed the games and wished to repeat corresponding activities. (b) They considered that by playing, they acquired basic skills and knowledge as far as programming principles were concerned. (c) They preferred Pair Programming and estimated that they could learn better by assisting one another. Finally, it seems that the few hours visit was not enough to motivate most pupils to carry on with the activities at home by themselves.

This chapter is organized as follows. In Section 7.1, we present background of research study. In Section 7.2, we present the European Code Week effort. Following, in Section 7.3, we discuss the methodology that was adapted in this study. The results of the study are presented in section 7.4, while Section 7.5 summarizes the most important conclusions. Finally, section 7.6 presents the elements that this study enhanced during the design of the ATMF.

7.1 Introduction

In the context of the European Code Week of 2014[40], pupils of 5th and 6th grade of Greek primary schools visited a public university and with the help of undergraduate computing students, they played digital educational games from code.org website and specifically the activity "K-8 Introduction to Computer Science"[41] which was designed for the global "Hour of Code"[42]. After having finished with the computer gaming activities the pupils answered a brief questionnaire with the purpose to discover their perceptions about GBL, CS and PP. Furthermore, the research examines whether a rather short-term action that is to say a brief visit to the university[43], can result in long-term motivation of pupils in order to continue the digital gaming activities at home.

Code.org website hosts a lot of "serious" games given the fact that they are not purely entertaining but mainly educational [624]. In our research we made the most of our gaming activities using the lesson entitled "K-8 Intro to Computer Science", which was actually the most popular of the action. Every activity-game hosted at the Code.org website is free. The particular game series aims at

[39] Parts of this study have been published in: Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2015). The Little ones, the Big ones and the Code: Utilization of digital educational games in primary school pupils. In *Proceedings of the 7th Conference on Informatics in Education (7th CIE2015),* Piraeus, Greece, 49-59.
[40] Europe Code Week: http://codeweek.eu, last accessed January 2017
[41] Teach the K8 Intro to CS: https://code.org/educate/20hr, last accessed January 2017
[42] Hour of Code: http://hourofcode.com/gr, last accessed January 2017
[43] The event at the EU Code Week site: http://goo.gl/klxetg, last accessed January 2017

demonstrating to pupils the idea that programming can be fun as well as to demystify CS. During the activities, pupils learn to cooperate and be creative. The activities do not require prior knowledge nor dexterity of programming. The games in question also strengthen the ability of problem solving and therefore have been developed according to the international educational standards[44]. As far as the programming field is concerned, all activities are based on VLPs. Pupils are able to learn programming without bearing in mind how to use syntax, but simply by focusing on the rational of how to write programs.

7.2 European Code Week

The European Code Week was held for second consecutive year between 11 and 17 October 2014. Over 150.000 children, parents, teachers, businessmen and politicians participated in events and took part in seminars on how to acquire basic programming principles and interrelated skills. Code Week aims at familiarizing the public with programming principles, providing opportunities for learning and bringing people with similar interests in contact with one another. According to digital agenda for Europe[45] computer science and programming promote computational thinking and according to Wing [67] they consist basic skills for the 21st century.

In Greece, the CS school counselors under the auspices of the Ministry of Education[46], supported and encouraged CS teachers to participate in the European code week[47] (Figure 69). Consequently, there were plenty of primary and secondary education teachers who volunteered in this call and in turns put some activities for the promotion of learning and programming into action. The activities held in Greece for 2014 surpassed 460[48] ranking the country among those with the most actions in a European level.

Figure 69. The Greek poster of CS teachers' counselors.

[44] Common Core NGSS: http://goo.gl/zj1D5o and CSTA standards: https://goo.gl/5kirUK, last accessed January 2017
[45] Digital Agenda for Europe: http://ec.europa.eu/digital-agenda/en/coding-21st-century-skill, last accessed January 2017
[46] Greek Ministry of Education ruling about EU Code Week: 148023/Γ2 /17-09-2014
[47] CS Counselors website about Code Week: https://cseduweek.wordpress.com/, last accessed January 2017
[48] Greek events on EU Code Week site: http://goo.gl/xvwHCi, last accessed January 2017

7.3 Methodology

In the present research, 94 pupils from public primary schools participated and paid a visit to the University of Peloponnese and particularly to the Department of Informatics and Telecommunications (Figure 70). The participants were between 10 and 11 years old attending 5th and 6th grade of the primary school then, and through this action, they had the chance to acquire their very first experience with programming. The venue took place at the modern computer laboratories of the department at the University.

Figure 70. The university's poster about the coding event.

Initially, the pupils were presented with a short video relevant to the value of programming and CS. Next, the young pupils met the elder ones, the undergraduate volunteers who were members of the students' department branches of IEEE and ACM [49]. The undergraduates informed the young visitors about the department and CS and answered all questions.

Afterwards, the more experienced programmers led the newcomers to their first steps with the code, via educational environments, which were game-based. At first, they assisted children to create electronic accounts on the Code.org webpage in order to make them able to move on with their activities even when they would leave the university. For this reason, pupils kept records with every account data on a piece of paper, which was the same for both of the two group members. On this sheet, one could also read a short description of the purposes and activities of the European Code Week, in order to inform the pupils' parents.

After creating personal accounts, the pupils could start playing. This stage lasted for about 2 hours. Pupils worked in pairs (Figure 71) which were randomly selected and in some circumstances coordinated by their teachers' recommendations. Therefore, there were 47 pupil-pairs. This way of working known as Pair Programming (PP) and pupils get advantages as they help each other [236, 625, 626]. PP also allows pupils to comprehend that CS can be societal and cooperative [627, 628].

Figure 71. Pupils working together.

[49] Student chapter of ACM, University of Peloponnese: http://uop-acm.blogspot.gr, last accessed January 2017

To increase working efficiency between group members, the following rules were given:

- One pupil in each pair would have the guide's role, who controls the mouse and keyboard.
- The other would have the navigator's role, who makes suggestions.
- Pupils were obliged to reverse the above roles at least twice.

During the entire process, the undergraduate pupils played the role of coordinators, who motivated and assisted the young pupils whenever there experienced a difficulty (Figure 72). Young and elder cooperation was extremely fascinating.

Figure 72. Undergraduate students help the little pupils.

When the activities came to an end, pupils filled in an anonymous questionnaire, containing close ended questions in order either to confirm or refute the research hypotheses (H1-H2-H3). The answers were based on the Likert's scale and ranged between 1 (*surely not*) to 5 (*surely yes*) containing intermediate values which indicated relevant grading, as well as value 3 indicating a neutral attitude.

Pupils were also awarded a participation certificate by the University of Peloponnese, which demonstrated with great happiness and pride (Figure 73).

Figure 73. Pupils showing their certificates.

In the present research, four main hypotheses were studied:

Hypothesis 1 (H1) - Pupils would be interested in programming activities and would find them very important and relevant for their lives.

Hypothesis 2 (H2) - Pupils would find digital games appropriate for the teaching of programming. In particular, H2 depended on the following questions:

- Whether they liked the activities.
- Whether they believe they have learnt some basic programming principles.
- Whether they wish similar actions to be repeated in future.
- Whether they believe they have higher motivation in learning, by playing digitally.

Hypothesis (H3) – pupils would prefer to work in pairs while programming and have the sense that they learn better, when they cooperate.

Hypothesis (H4) – Can a short duration programming activity motivate the pupils to continue programming after the end of the session? As far as the H4 is concerned, the researchers used the teacher's dashboard application, where the pupils' progress in gaming was recorded (Figure 74).

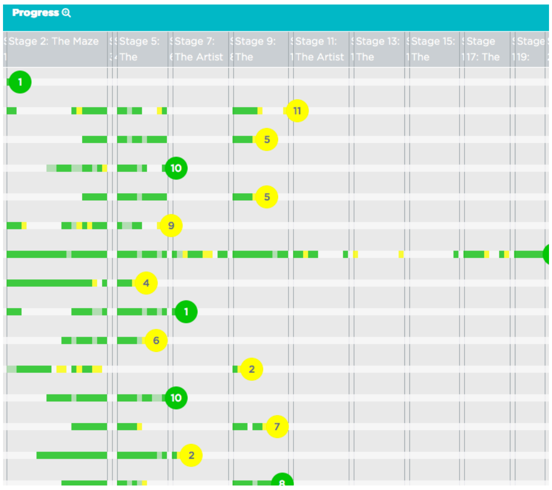


Figure 74. Teacher's dashboard showing pupils' process.

7.4 Results

In this section, the research findings are presented according to the hypotheses, as well as the purposes described above.

Computer Science and Programming (H1)

According to the pupils' answers, the greatest percentage (76.6%) thought that it was important to learn programming. More analytically, the 57.4% of pupils declared "*probably yes*" and the 19.1% "*surely yes*". On the contrary, just 1.1% (a single pupil) answered "*surely not*" and 13.8% answered "*probably not*". Similar results were found for the statement "*I want to learn more about the Computer Science*". The majority of 81.9% of pupils expressed a positive view and just the 9.5% of them a negative one. Thus, H1 was supported.

Digital Educational Games (H2)

The activities that have to do with the digital gaming received 86.2% of positive answers, while just the 5.3% declared "*probably not*". It is quite remarkable that no pupil declared "*surely not*".

Moreover, it is also highly noticeable that an amount of 60.9% believed that they had learnt some basic programming principles through games. Even if the figure is not too high, nevertheless the majority of pupils estimated that the games were effective. For the same question, the 8.7% declared "*surely not*" and finally the 16.3% (that is to say 15 pupils) had a neutral opinion.

In addition, most of the pupils expressed the desire for a repetition of similar actions in future (88.3%).

Finally, it is clear that the digital games give great motivation to pupils since 66% of them declared that they prefer learning through these. Therefore, H2 was supported.

Pair Programming (H3)

According to the pupils' answers, the majority (56.4%) preferred working together with a co-pupil on the computer for programming through games. Contrary to this, the 21.3% expressed a negative view but the amount of the neutral view is considered to be a high one (22.3%).

Similar to the above were the results to the statement "*I believe that I learn better when I work together with a fellow-pupil on the computer*". 59.6% of them had a positive view, the 23.4% a negative one, while 17% a neutral one. Thus, H3 was also supported, although it was not a strong positive majority of answers.

Long-term Motivation (H4)

On the code.org website via the automated dashboard, teachers can see some general statistics for each pupil account with the stages completed. The researchers connected to the administrative account three times with a week-interval between those times. During these three weeks, most of the pupils did not connect at all to their created accounts. The accounts were as many as the pairs of pupils (47 in total), meaning that there was one account for each pair, but the pupils individually could log in at a later stage and from a different location and play the games as they wished. Just 10.6% of the pair-accounts that were registered during the visit to university, connected at least once and tried to play and complete some stages more (it could be just one member or both connecting). On the contrary, the 87.2% did not carry since the accounts were not activated after the visit. Therefore, H4 was rejected.

7.5 Conclusions

In order to succeed in obtaining positive educational effects, one of the basic prerequisites is the motivation of the pupil. Especially in CS the full participation of the pupils is required. In this light, tools and methods need to be implemented in order to motivate the pupils and enhance their active participation. Digital games in education can be supportive towards this effort.

Through the bibliographical review, it was found that many researchers support their use as a subsidiary means for the attainment of the learning aims. As it was shown by the present research, games have an important effect on the young population and promote the cooperative spirit. Especially as far as programming is concerned, including digital gaming, the value of cooperation and learning from another pupil is very important and is getting more and more popular [230, 629].

However, even when the pupils declared they had a high motivation in order to learn through educational games, there were only few who actually continued, by doing so at home. Therefore, it seems that short duration actions are not capable by their own to motivate the pupils in the long run.

On the other hand, the incorporation of digital games at school environments can improve the school image in the eyes of the pupils [630] and could lead to long term effects.

Our future research will focus on the development of a theoretical framework for digital games for CS and their incorporation in formal education. Specifically, it will be a theoretical framework for the design and development of educational environments, taking into account the possible different pupils' characteristics, exploiting all these games features that are capable of encouraging and motivating, making the learning procedure effective. Learning programming from a young age can very well reinforce the pupils' success in any career of the 21st century and in this case, digital games could be the basic tool in this endeavor.

7.6 Elements in the ATMF

This section presents the elements of the ATMF that were affected by this research study (RS3). Figure 75 shows the methods that were used.

ATMs		
Element 2		**Teaching**
	Section 1	Peer Learning
	Section 2	Problem-Based Learning
	Section 3	Project-Based Learning
	Section 4	Studio-Based Learning
	Section 5	Inquiry-Based Learning
	Section 6	POGIL
	Section 7	Team Learning
	Section 8	Game-Based Learning
	Section 9	Educational Robotics
	Section 10	Non-Textual Programming
	Section 11	Contextualized Learning
	Section 12	CS Unplugged
	Section 13	Subgoal Learning
	Section 14	Programming Puzzles
	Section 15	Extreme Programming
	Section 16	Program Visualization
	Section 17	Competency-Based Learning
	Section 18	Social Networks Learning
	Section 19	Emerging Technologies

Figure 75. ATMs used in the RS3.

Figure 76 displays the highlighted elements of each component after RS3. From the first dimension and component 2a, the element about time restrictions is mentioned. The school's visit to the university held only two hours due to time restrictions. That seemed to affect the pupils' long term motivations. In addition, this study shows positive attitudes from young (element 1, component 2a, dimension 2) learners (element 7, component 2a, dimension 2).

From the third dimension, two elements are revealed in component 3a. Long term planning seems to be an important factor since as shown in this study did not have any long term motivations. This also affects the expectations that instructors must have when planning a lesson (element 5).

Finally, from the fourth dimension, three elements from component 4a were highlighted. Element 1 about the knowledge gained, element 2 about the students' motivation and element 3 about students' skills. From component 4b, the second element about monitoring of learning is mentioned, since such a process was used in order to evaluate the long term effects (element 5).

Dimension 1: Context		
Component 1a		Education Level
	Element 1	Formal Education / Type of School / Level
	Element 2	Informal Education
	Element 3	Non-formal Education
Component 1b		Environment (physical)
	Element 1	Infrastructure (availability)
	Element 2	Supporting Resources
Component 1c		Educational system
	Element 1	Policies
	Element 2	Compulsory Curriculum
	Element 3	CS standards
	Element 4	General public attitudes
	Element 5	Time restrictions

Dimension 3: Content		
Component 3a		Learning
	Element 1	Subject
	Element 2	Objectives
	Element 3	Short-term & Long-term Planning
	Element 4	Expectations for learning
	Element 5	Relations to other subjects / disciplines
Component 3b		Teaching
	Element 1	Lecture-based
	Element 2	ATMs (see Figure 33)

Dimension 2: Participants' Characteristics		
Component 2a		Learner related
	Element 1	Age
	Element 2	Gender
	Element 3	Cultural background
	Element 4	Socioeconomic status
	Element 5	ICT use
	Element 6	ICT experience
	Element 7	Attitudes
	Element 8	Personality traits
Component 2b		Instructor related
	Element 1	Age
	Element 2	Gender
	Element 3	Teaching experience
	Element 4	Academic characteristics
	Element 5	Skills - Abilities
	Element 6	Attitudes
	Element 7	Personality Traits

Dimension 4: Evaluation		
Component 4a		Learner related
	Element 1	Knowledge / Performance
	Element 2	Motivation
	Element 3	Skills
	Element 4	Metacognition
	Element 5	Beliefs
Component 4b		Instructor related
	Element 1	Assessment Criteria / Method
	Element 2	Monitoring of Learning (Learning Analytics)
	Element 3	Feedback to Learners
	Element 4	Long-term Evaluation
	Element 5	Teaching Evaluation

Figure 76. ATMF Elements of RS3.

7.7 Summary

This research study involved elementary pupils who worked in pairs trying to solve programming puzzle games. Participants reported satisfaction and willingness to use GBL while they estimate that they acquired basic programming knowledge. They preferred Pair Programming and also estimated that they could learn better by assisting one another. Furthermore, this short activity like did not had long term motivation at pupils.

Chapter 8

The goal is to teach in such a way as to produce the most learning from the least teaching.
Seymour Papert

Research Study 4: Educational Robotics

The present chapter [50] is an observational study recording the participating teachers' attitudes from the PanHellenic Educational Robotics (ER) competition. Teachers' opinions towards ER, regarding their students' involvement with robotics, during the preparation for the competition are reported. The study (RS4) investigates the benefits of students' involvement with robotics about skills, motivation and learning. Additionally, it is researched weather ER should be introduced in the compulsory curricula. A qualitative methodology was used, through questionnaires with open-ended questions. Although the sample was relatively small (*N=18*), the results were quite homogeneous showing a very high level of engagement and motivation of teachers and students. The results show that there are numerous benefits for students: they seem to increase their collaboration, problem solving and creativity skills; understand STEM concepts about computer science and engineering and especially gaining programming knowledge. In addition, most of the teachers consider that ER should be part of the compulsory curriculum. The effectiveness of robotics, through the eyes of a national competition, shows that the robotic activities are more challenging than lack of resources, time and assessment issues, for both students and teachers. ER should be an essential part of the school and through challenging competitions and activities, it can bring together young people from all over the world to learn and develop important skills for the 21st century.

This research study is structured as follows: Sections 8.1 and 8.2 provide the background of the study with a review of related literature. Section 8.3 deals with the research questions and the context of the study adopting the Darmstadt model. Next, at section 8.4 (methodology), the procedure and design, the participants and applied instrumentation of the study are described. The findings of the study are presented in section 8.5 whereas they are discussed in section 8.6. Section 8.7 is about the conclusions, limitations and future work. Finally, we close with section 8.8 about the contribution of this study to the ATMF.

8.1 Introduction

Why introduce Educational Robotics (ER) in formal school education? A possible answer could be that early involvement with ER will increase students' motivation for STEM (Science, Technology, Engineering and Mathematics) related education and will make CSE more attractive to students. Successful participation in ER activities at an early age should be able to improve students' self-efficacy and therefore increase their motivation to attempt CS and STEM studies at the secondary and even tertiary levels [293].

[50] Parts of this study have been published in: Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2016). Educational Robotics in the Service of CSE: A Study Based on the PanHellenic Competition. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (WiPSCE '16), Münster, Germany. ACM, New York, NY, USA, 84-87. *DOI*=http://dx.doi.org/10.1145/2978249.2978262

However, the most important in education is learning. As educators, we want to know the learning results too. Successful learning is not a totally separate issue from motivation and skills, since it is hard to see them apart. The present work collected qualitative data regarding ER in schools, based on teachers' opinions and ideas after their engagement with the PanHellenic robotics competition. Our goals in this research are to: (a) focus on students' involvement with robotics during their preparation for the national competition, (b) perform a qualitative research: focus on teachers' attitudes; (c) investigate different aspects of skills, learning and motivation. If we could show that learning ER can help learn CS principles at school level, it would amplify the justification for need of ER in the compulsory curricula.

This study is structured as follows: Section 8.2 provides the background of our study with a review of related literature. Section 8.3 deals with the research questions and the context of the study adopting the Darmstadt model. Next, at section 8.4 (methodology), the procedure and design, the participants and applied instrumentation of the study are described. The findings of the study are presented in section 8.5 whereas they are discussed in section 8.6. Finally, we close with section 8.7 about conclusions, limitations and future work.

8.2 Background

ER and CSE

Many types of approaches and activities have been proposed for introducing CS principles and concepts to children [100, 631, 632]. As Armoni et al. [633] report in their study, activities in CS can fall into three main categories: 1) Kinesthetic activities, that do not use a computer, 2) Visual programming environments, where students create programs by using simple interfaces, and 3) Robotics, activities that are widely used to introduce science and technology to young students.

In this study, we describe our research with the ER, a teaching method that relies on the use of robots for teaching purposes [290]. As Rockland, et al. [293] underline, there is need to bridge the gap between the curriculum and the scientific nature of STEM-related concepts, by engaging students with alternative attractive learning approaches.

ER and competitions

Most robotics in education can be found in the middle and high school levels where over the last years, a number of competitions are available and many students participate. These competitions provide educational settings that increase motivation [634], engagement and even training in specific domains like Human-Robot interaction [635]. Their use has grown in popularity as demonstrated by the explosion of national and international robotics competitions, such as: World Robot Olympiad (WRO™), FIRST Robotics Competition (FRC), Boosting Engineering, Science, and Technology (BEST), VEX Robotics (VEX), and First LEGO League (FLL).

Characteristically, many of the robotics courses offered at schools all over the world nowadays, are concerned with ER student competitions [636]. Previous study from Melchior et al. [637] evaluated the impact of FRC on secondary school students. The results showed improved students' interest for STEM and increased self-confidence skills and problem solving abilities. Another study [638] provides evidence that there is strong need for the continued growth and expansion of ER competitions like the FIRST, particularly into communities serving low income and minority youth.

The present paper is built upon the PanHellenic robotics competition of WRO™ Hellas (Organization for Educational Robotics, Science, Technology and Mathematics). Previous works focus on the evaluation of competitions investigating the long-term impact of robotics on former participants by the students' perspective [638, 639]. Our study focuses purely on teachers' perspective in addressing questions about robotic education, trying to fill the gap around the much needed information regarding CS teachers' motivations and experience. Diethelm et al. [492], emphasize the importance of the teachers' perspective since a topic can be reached by many different levels.

Finally, there are certain factors that affect students' motivation to engage with ER activities like easiness of use and perceived usefulness together with the intensity of the educational experience they offer [640]. In this light, ER competitions seem to offer the required elements for intensifying the experience and thus provide learning motivation. However, this claim remains to be confirmed in the present work.

8.3 The Research

Research Questions

For the purposes of our research, we posed the following three research questions:

RQ1: What are the benefits of students' involvement with ER through a competition, regarding skills and motivation, as teachers perceive them?

RQ2: Can ER be used to teach basic STEM principles like programming and engineering, in Greek education?

RQ3: Should ER be introduced in the Greek compulsory curricula? And if so to what level?

Research Context

Recently there has been an attempt to create a holistic framework for CSE in schools from different cultural backgrounds. The developed framework aims at presenting the diverse factors that affect CSE, in order to be able to compare findings from different areas and transfer findings. This framework is known as the Darmstadt model and is presented in Hubwieser [641]. Within the framework of the Darmstadt model, Greece was represented as an extensive case study showing practices regarding CSE in Greek education [642]. We wish to approach the issue of educational robotics within that framework in order to provide additional information. However, a complete analysis of the Darmstadt model is outside the scope of the present work and for this reason, only a short description will follow. In this observational study, questionnaires were collected from teachers/coaches that participated in the PanHellenic educational robotics competition 2015 of WRO™ Hellas on 3 October 2015[51]. The Range of Influence is, therefore, covering the whole of Greece including primary and secondary education. The Educational Relevant Areas axes are as follows:

Socio-cultural factors: CS courses are not considered as core courses in Greek primary and secondary education both from the view of the society and the curricula. The engagement with Educational Robotics is an extracurricular activity.

[51] WRO Hellas PanHellenic competition (in Greek): http://goo.gl/RmyIXX, last accessed on April 2016

Policies: Due to the economic crisis in Greece, there is lack of CS teachers and laboratory infrastructure in many cases mostly in provisional schools. Educational Robotics require financial resources that in many cases are not available (will be described below).

Teacher Qualifications: CS teachers are highly qualified with University degrees. The participating sample is described in the method section in detail.

Curriculum Issues: CS courses are taught only one hour per week in grades seven to nine. In tenth grade (first year of senior high school in Greece), CS is an optional course. In eleventh grade, is taught an hour per week and finally in the last year of secondary education, is taught just in specific scientific fields. Educational robotics are not a part of the schools' CS curriculum. In Greek primary education CS is taught in selected schools with reformed curriculum (called EAEP: FEK 804/2010).

Examination/Certification: CS courses in Greek primary and secondary education do not lead to any certifications.

Motivation: In Greek educational system, students finishing the secondary education, take national exams in order to enter higher education. A programming course is prerequisite to enter economic and computing universities.

Teaching methods: As far as curricular activities concerning CSE in Greek schools, teaching usually follows traditional patterns like lectures, fixed exercises and exams. Educational robotics are a hands-on experience. In the case of Greece, all activities happen outside school hours and on a purely voluntary basis from all participants (students and teachers).

8.4 Methodology

The main purpose of this study is to investigate the impact of robotics in education, and in particular LEGO® Mindsotrms®, on students' learning, motivation and skills. Furthermore, the study intends to present Greek teachers' attitudes towards robotics in education.

Procedure and design

The WRO™ Hellas carried out for 7th year the PanHellenic educational robotics competition, which takes place during the annual Olympiad organized by the international World Robot Olympiad. The competition held at Athens in October 2015 under the auspices of the Ministry of Education and there were over 300 students participating from all over the Greece.

The competition followed the official rules conducted by the WRO™ Hellas, in order to distinguish the teams that will travel to the Olympics. In order a team to join the competition it should have passed through the regional competitions that were organized through all over the country between May and September 2015. The theme of the competition was, as in the Olympics, the "Robotic explorers", under which students were asked to build a robot to explore different environments (e.g. seabed for elementary schools or the mountains for the senior high schools).

As for the robots that were used, WRO™ allows only the use of the LEGO® Mindstorms® robotics platform for WRO™ Regular Category (Elementary, Junior High, High School), and this policy is in place to ensure equal opportunities for all participants. The competitions aim to challenge the participants to use their creativity to construct and program autonomous robots and the LEGO® Mindstorms® platform is found to be the most appropriate.

An online survey was carried out after the competition, in November 2015. The survey was targeting teachers that participated in the PanHellenic ER competition of WRO™, although invitations were extended widely. It was published via the WRO™ Hellas forum, as well as through social media channels. In order to address the research questions this study relied on qualitative data collected by the questionnaire.

Participation

All participants were teachers leading different competing groups at the PanHellenic Educational Robotics competition of WRO™ Hellas. Each teacher could be the instructor-coach of one or more teams. Eighteen completed questionnaires were received. Table 31 lists the profile and expertise of the 18 participants, from whom *N=6* (33.3%) were females and *N=12* (66.7%) were males. The majority of the participants had a higher education IT background and most of them had a long teaching experience.

Table 31. Profile of the 18 participant teachers in the ER competition.

Teacher	Specialty	Groups	Experience
T1	CS	5	> 10 years
T2	CS	2	> 5 years
T3	Mathematician	1	> 10 years
T4	CS	4	> 10 years
T5	Primary Teacher	1	> 10 years
T6	CS	3	> 10 years
T7	Science/CS	1	> 10 years
T8	CS	5	> 5 years
T9	CS	5	> 10 years
T10	CS	4	> 10 years
T11	Primary Teacher	4	> 10 years
T12	CS	1	> 10 years
T13	CS	1	> 5 years
T14	Science	5	> 10 years
T15	CS	3	> 10 years
T16	CS	5	> 5 years
T17	Electrician	2	> 10 years
T18	CS	2	> 10 years

Most instructors participated in the competition with more than one team. It seems that teachers are highly motivated to participate in such activities, since each team's preparation requires many resources and personal time from the instructor.

In addition, most teachers met a least once a week with their students in order to practice with robots. Most of the meetings took place in schoolrooms like the computing labs or the school libraries.

Instrumentation

The instrument for assessing teachers' attitudes/opinions was a 22 items mostly open-ended questionnaire (OEQ) but also including some multiple-choice questions (MCQ). The questionnaire combines different standardized assessment tools as well as survey instruments, which have been validated and/or applied in previous studies [643-646]. The questionnaire is structured around five main sections (Parts) and is divided into relevant sub-sections (from a to d) that describe briefly the questions:

Part A: Demographic/background information (5 items; MCQ)

(a) Confounding factors: teaching experience, teacher specialty.

(b) Statistical information: age, gender, school.

Part B: Competition Preparation (8 items; MCQ/OEQ)

(a) Funding resources: money for the robots, supplies and travel expenses.

(b) Difficulties: problems that teachers faced during the preparation time for the national ER competition.

(c) Instructor-coach: number of participating teams, members.

(d) Meetings with the students: how often, when and where.

Part C: Students' Motivation/skills (2 items; OEQ)

(a) Motivation: what motivated students most from their involvement with robotics.

(b) Skills: skills that students acquired from ER.

Part D: Students' Evaluation (2 items; OEQ)

(a) Programming: what did students learn as far as programming is concerned.

(b) Engineering: what did students learn as far as engineering is concerned.

Part E: Attitudes (5 items; MCQ, OEQ)

(a) Curriculum: ER in compulsory education, to what level, STEM education.

(b) Society: parents' attitudes towards ER, technological convergence.

(c) Open source robots: robots based on open source software, advantages, disadvantages, other robotics competitions.

Data Collection and Analysis

The data collected were mostly from open-ended question from teachers' responses to our questionnaire. The questions were designed to investigate different aspects of learning, motivation and creativity. The sampling process was done randomly from online media rather than making a specific selection amongst the teachers. However, the sample was sufficient for the purposes of the study due to its qualitative nature.

We gathered responses from 18 questionnaires but it was clear that we had reached to a satisfactory point of saturation: gathering more answers was not expected to provide radically different data, since we could already recognizer patterns. Therefore, the qualitative data were coded in an inductive manner with respect to emerging themes, following the guidelines in [647]. The themes were then grouped to facilitate further analysis. Thus, a set of codes was developed shown to ensure agreement on the interpretation of the teachers' statements. The coding scheme that was used, in correspondence with the parts of the questionnaire, is presented:

Part A: Demographic/background information

No codes were used – only MCQ questions

Part B: Competition Preparation

(a) Funding resources: schools, donations, EU projects, parents' association, own expenses.

(b) Difficulties: lack of money, lack of time, coordination issues, dropouts.

Part C: Students' motivation/skills

(a) Motivation: challenges, collaboration - team working, relation to real world, creativity, self-confidence, enjoyment.

(b) Skills: problem solving, cooperation and task management, creativity, sociability, discipline.

Part D: Students' Evaluation

(a) Programming: basic principles, structural programming, visual programming, computational thinking.

(b) Engineering: basic principles, construction-assembling, action-reaction principles, forces and torques concepts.

Part E: Attitudes

(a) Curriculum: compulsory education, from elementary education, from secondary education, in all levels, STEM education.

(b) Society: parents' attitudes towards ER, technological convergence.

(c) Open source robots: positive attitude, money issues, WRO™ competition, better documentation, faster growth.

8.5 Findings

In this section, we describe our findings: a) regarding the students' involvement in the competition, about skills and motivation (RQ1), b) regarding STEM aspects that are related to the learning evaluation (RQ2), c) about robotics in compulsory education from teachers' attitudes (RQ3) and, d) some additional findings like difficulties that the teachers faced during their preparation for the competition, in order to enhance the results and shed light on some of their aspects. Although the sample was relatively small (*N=18*), the results were quite homogeneous and thus it allowed us to present the findings in a grouped approach.

ER - motivation and skills acquired

As far as motivation from students' involvement with robotics is concerned (Table 32), all teachers are convinced that students enjoyed the robotics and that is the biggest motivation. Characteristic, teacher T17 replied:

T17: I think the chief benefit of educational robotics is the motivation that acquire children, for learning by doing in a very pleasure way.

Furthermore, many teachers responded that students were motivated by the novelty and challenge that robots offer (8 cases). Relatively, teacher T2 wrote:

T2: Students liked most the fact that they dealt with something innovative and challenging through the competition. They were motivated by the fact that they could create something from nothing.

Some teachers also believed that students were motivated by the collaborative nature of the activities during the preparation time for the competition (4 cases). Certain teachers' responses were:

T4: They were excited with the value of teamwork, a conception that even it is taught through a related course in schools (named Project), they could not understand.

T13: Students enjoyed working in groups with their classmates. Collaboration and mutual support are essential ingredients for such activities. Actually, my students collaborated with harmony and in such a way that I did not expect.

Additionally, the fact that ER offers direct outcomes and students can directly see the result of their activities, was mentioned 3 times (Table 32). Teacher T15 responded:

T15: My students were motivated to use robots by the fact that they had direct feedback of their actions and the possibility of immediate correction in correlation to real world problems.

Yet, three more teachers thought that students were motivated by the nature of robotics, since they participated in a highly creative process. Teacher T18 mentioned:

T18: Students were inspired by the challenge of seeking a creative solution to a complex problem.

Other responses were also mentioned like:

T4: The involvement with robotics made students feel special and different in a positive way than other students.

T12: … the possibility to travel to other places in order to participate to the international competition; and the competitive atmosphere that was created for the national competition.

Table 32. Motivation: most commonly occurring themes.

Motivation	**Number of**
Enjoyment	18
Novelty/Challenge	8
Collaboration-Team working	4
Relation to real world	3
Creativity	3

Next, as far as skills is concerned, Table 33 summarizes the grouped results from teachers' opinions. Teachers consider that most students can develop numerous skills from their involvement in such activities. Moreover, everybody mentioned that students improved important skills such as: problem-solving, collaboration and project management. The responses are justified nicely in the words of the teachers:

T11: It was impressive to see primary school students implementing solutions by breaking a problem into sub-problems and trying to find solutions to simpler tasks.

T4: I think the most valuable skills that the students learned was the essence of the team spirit and the need for collaboration. There were tasks that even the better students did not manage to accomplish and the solutions came from collaboration with their teammates.

T3: The participation in the national competition was a huge project itself. My students dealt with matters about planning and monitoring the project to all its phases. Together we shaped the goals, we found resources and we created schedules for each phase.

Moreover, most teachers thought that students developed creativity skills, since such responses were mentioned *N=10* times from different instructors. One of them made the following remark:

T12: The building part of the robot (assembling the LEGO® blocks), required solutions that even me could not think. Students came up with better ideas many times and thus I think that one of the skills that they developed was the creative thinking.

Additionally, *N=7* teachers estimated that students increased their sociability skills. Teacher's T4 answer about social skills was representative:

T4: Even more introvert students developed communication and interaction skills after some time.

Finally, some teachers provided extra skills they thought the students gained, as an addition to the above. Their responses included increased self-esteem (*N=3* cases), computational thinking (*N=3* cases) and discipline skills (*N=2* cases).

Table 33. Skills: most commonly occurring themes.

Skills	Number of cases
Team spirit	18
Problem solving	18
Project management	18
Creative thinking	10
Sociability	7
CT	3
Self esteem	3
Discipline	2

ER - learning STEM concepts

In order to examine if robotics can be used as an alternative to teach CS and STEM concepts, two parts were evaluated: programming and engineering. Those were the two disciplines that the students got involved with the robots.

As far as programming is concerned (Table 34), most teachers (*N=18*), responded that students grasped programming concepts more easily and understood structural programming concepts because of the visual programming language (EV3) that the robots use. This view is demonstrated here from teacher's T1 response:

T1: With the visual programming students understood many programming concepts easily, like variables, loops, switch, etc. They could see things happen and follow up the logic of a program and consequently understand the need to implement structured programs.

Additionally, most teachers (*N=14*) reported that students learnt basic programming principles. In particular, teachers mentioned that robotics helped their students understand basic principles such as variables, conditions and loops. Most also created functions with parameters and used arrays in their programs. This impression is analytically shared by teachers T7, T13 and T18:

T7: Students understood very easily the meaning of variables by changing them, comparing them, looping over them and returning their values in one way or another. Overall, they did not learn anything too confusing, just the steps to make things work. However, let's just say that they can create basic programs on their own by now while they had no previous programming experience.

T13: Robotics is an excellent way to teach repetition structures. The tasks we had to accomplish required to repeatedly processing one or more instructions until some condition is met. Most tasks were repetitive, and seeing the robot to do what the program said was the best way for students to understand the concept.

T18: My students learned structural programming by the meaning of functions and using them when needed and by changing their input. They also understood the "bug" conception, since many times their programs had logical errors.

From all the aforementioned, it seems that all teachers (*N=18)* consider that students understood the "divide and conquer" idea of structural programming.

Finally, most teachers (*N=12*) thought that students acquired CT skills. Although CT was found in teachers' answers about skills, it also appeared in many cases of the evaluation part. Teacher T3 made the following remark about the programming learning:

T3: Students became familiar with basic algorithmic structures (sequence, selection, and repetition), used variables, logical conditions, nested structures, created procedures with parameters and processed dimensional arrays. They learned to decompose the entire decision making process, and tried to find all possible solutions, ensuring that the correct decision was made based on the corresponding parameters and limitations of each problem. In other words, they seem to have followed what we call CT process.

Table 34. Programming: most commonly occurring themes.

Programming Evaluation	**Number of cases**
Structural Programming	18
Visual Programming	15
Basic Principles	14
CT	12

As far as engineering is concerned, 9 teachers believe that students learnt basic engineering principles (Table 35). More specifically, *N=5* teachers state that their students used the concepts of forces - torque, and friction. Furthermore, *N=4* teachers consider that students acquired action and reaction principles like the use of gears, acceleration and deceleration and the motion transfer from horizontal to vertical axis. The words of teachers justify their thoughts:

T6: I believe that by the end of the competition my students were able to describe basic engineering concepts that had to do with the robot structure and movement: they understood the relationship between work, force and power; they used gears, acceleration and deceleration concepts; they tried torque in different settings.

T14: We had to make our robot more flexible to reach the mountains (one of the tasks) from distance. We used all the possible hardware we could, like motors, sensors and gears but it was also necessary for students to search for better solutions by using their knowledge from physics, like forces and torque and action-reaction principles.

T17: A big part of the competition had to do with robot construction to accomplish faster and more efficient the given tasks. At that point students had to practice with engineering concepts (forces and torque as well as action and reaction principles) to find the best structure.

Additionally, most of the teachers (*N=12*) supposed that students learned solving construction and assembling matters through the building blocks and the different constructions they had to made. Teacher T5 made the following remark:

T5: Children are familiar with LEGO® blocks and use them so easily to create robot solutions that are outside our view.

Nevertheless, the discipline of engineering is extremely broad, and encompasses a range of more specialized fields. To that point of view is worthwhile to mention teacher's T3 response about not encouraging students to learn engineering concepts:

T3: There were so many things I wanted to give to the children, where I unwittingly or not passed through the prism of CS. If I had more time, I could give more explanations using physical-mechanical and interdisciplinary approaches to the subject. Unfortunately, too much time pressure to catch up dates did not give me the ease to explain it so.

Table 35. Engineering: most commonly occurring themes.

Engineering Evaluation	Number of cases
Constructions-Assembling	12
Basic principles	9
Forces and torques concepts	5
Action-reaction principles	4

ER - curricula and attitudes

In order to examine if there is a place for ER in Greek educational system (RQ3), we gathered teachers' responses: enter the compulsory curricula and if so to what level; society views; other than LEGO® robot choices.

As far as curricula and ER is concerned, it was not surprising that the majority of the teachers (14 out of 18) think that robotics should be a part of the compulsory curriculum (Table 36). Characteristically, teacher T9 reported:

T9: Let's look around us in other countries, not to copy their programs blindly, but to ask why we are so far behind. Robotics is a basic course in many educational systems from primary school to university and then is inextricably linked with the production-industry. At present in our country we rely on teachers' will who overcome themselves due to lack of structures and resources.

However, although the majority of teachers assume that ER is essential at schools, it seems that there are many different opinions about the level that it should be introduced. *N=5* instructors would like to see ER in all levels of education, including primary and secondary. Nevertheless, there were some concerns regarding very young students and *N=4* teachers thought that activities like these should be introduced after the 4th or 5th grade of the primary school. This impression was shared by teacher T11:

T11: The educational robotics should be taught from the elementary education. Students can learn at a young age cooperation and team spirit. My view is to start at 4th grade, because children of that age have the maturity for such educational activities.

Finally, it is worth mention the view of teacher T3, who supposed that robotics should not be considered as a separate discipline:

T3: ER do not need to be introduced as a separate topic in schools but can be a part of an interdisciplinary approach in the teaching of different modules like mathematics and natural sciences:

Table 36. Curriculum: most commonly occurring themes.

Robotics in curriculum	Number of cases
Compulsory curriculum	14
In all levels	5
Secondary education	5
Elementary education	4

Following, teachers responded to the question if their students' parents (referring to the parents of the participating students) have positive attitudes regarding ER. Everybody said that the parents are very positive when their children are involved in such activities. This is not a surprising finding, since their children were already participating in the activities. Nevertheless, it seems that the parents' positive attitudes remain, even after the long period of engagement of their children with robotics (during holidays, after school hours, covering expenses).

In addition, most teachers ($N=13$) think that ER is crucial for the technological convergence of Greece with the more technologically advanced countries. Teacher T8 replied:

T8: The most technologically advanced countries in the world have integrated their scientific advancements into the society, making life easier for citizens and increasing productivity. I believe that the best field to succeed this is education. Hence, ER could have a huge impact on the country by achieving excellence in the field.

Finally, teachers were asked to provide their opinion regarding other robots like those that use open source software. The results here (Table 37) were very interesting since most teachers would or have already used other robots based on open source ($N=13$). The most important factor seems to be economic ($N=10$ teachers), since such robots are cheaper than LEGO® Mindstorms®. Some teachers' answers are presented:

T5: There is really "no excuse" not to use robots since it could be done with very little financial support with open source platforms.

T11: I would use anything open source and "out of principle". One teacher said that both approaches have advantages and disadvantages and the main decisive factor is cost.

Yet, some teachers ($N=4$) believe that open source robots (e.g. Arduino) have better documentation and $N=3$ teachers believe that they have faster growth. Despite all the positive comments towards the use of open source robots, some concerns were also raised. For example, teacher T3 responded:

T3: I do not have the knowledge to support open source robots. I think that they require deeper programming knowledge and might not be appropriate for school students.

In addition, the robotics Olympiad from WRO™ seems to be a powerful motive for some (*N=4*) teachers. Teacher T2 made the following remark:

T2: Open source robots are appropriate for class activities but not appropriate for competition, since LEGO® are simply to use and good quality products.

Table 37. Other robots: most commonly occurring themes.

Open source robots	Number of cases
Positive attitude	13
Money issues	10
WRO™ competition	4
Better documentation	4
Faster growth	3

Additional findings

There were also various significant findings that although are not directly related to the research questions of the study, nevertheless they provide important information to support them. Those findings come from some extra questions that teachers had to answer about the national ER competition.

Firstly, we found some gender differences. More specifically, half of the teachers (*N=9*) mentioned that boys are more interested in robotics than girls (6 were neutral and only 3 disagreed).

Next, we asked about the funding resources of the teams. The educational robots are not provided by the Greek public schools and should be bought privately. As shown at Table 38, seven instructors managed to get expenses covered from school budgets. Many asked the parents association to offer money (*N=5 cases*). Some stated that they found sponsorships from municipalities and private companies and they also involved students in fund raising activities (*N=4 cases*). However, in 6 cases students, teachers and the school's principal covered the expenses from their private budgets.

Table 38. Funding resources: most commonly occurring themes.

Funding Resources	Number of cases
School	7
Donations	4
European Projects	1
Parents' Association	5
Own expenses	6

Participants were also asked to explain difficulties they might have faced with their preparation for the national ER competition (Table 39). *N=6* instructors said that they did not face any problems. N=3 instructors mentioned the financial problems they faced and *N=4* more talked about the timely efforts as main problems. Other problems mentioned was issues in coordinating the students, the nature of the task that was new to the students, lack of proper space and equipment and lack of information regarding the competition. We present here some of the teachers' answers:

T6: The biggest problem we faced was the lack of money. The competition required more equipment than originally purchased.

T12: It was very difficult to find sponsors to cover the costs of the Robotics Olympiad. Until the last minute, we did not know if we will be able to participate.

T14: Very difficult to coordinate groups. Time was an important issue, since most students had daily lessons outside schools, even during the summer months and their time was limited."

T17: Time was an issue for us. Children could not spend many hours for the preparation to the national competition. Some of them left due to time, since they had to focus on the upcoming years' national exams for universities.

Table 39. Difficulties: most commonly occurring themes.

Difficulties	**Number of cases**
No problems	6
Drop outs	5
Lack of time	5
Lack of money	3

Moreover, asking whether the possibility to win in the national ER competition provided adequate motivation for participation, most instructors (*N=13*) provided a positive answer (3 were neutral and only 2 negative). Finally, 15 out of 18 teachers would happily participate in similar competitions in the future (3 did not know yet).

8.6 Discussion

The first research question asked about the teachers' attitudes, concerning the students' benefits from their involvement with ER through a national competition regarding: skills and motivation. The results seem very promising. All teachers agreed that ER highly motivate students and boosts their will for learning. Teachers also noted their students to develop very important skills like problem solving, collaboration and creativity.

Analyzing further some of the findings, someone should note here that the level of engagement with ER for all the participants (students and teachers) was very high, indicating personal involvement for purely voluntary activities. Since most instructors coached more than one team (even more than 4 in some cases), the number of students implied here is quite large. The activities with the robots took place outside school hours and the preparation for the competition was not in a continuous time frame, since it was in October and 2.5 months of summer holiday had interfered. One could expect that this long break would make students lose interest, however, this was not the case. Being committed over such a long period, indicates high motivational levels from all involved participants, both students and teachers. This could be due to the very nature of educational robotics that seem to excite the student, but it could be also due to the upcoming competition which also motivated students. It is important also to note, that teachers did not have any financial benefits for their involvement with the ER and in many cases they had to contribute from their own.

The second research question asked about the teachers' attitudes, concerning the evaluation of students' involvement with robotics through the competition and therefore if ER could be used to teach basic STEM principles, like programming and engineering, at school level. Most teachers believe that their students can learn STEM principles like programming and engineering with robotics. Although robotics does not focus exclusive on programming, there seems to be the main learning outcome like previous study has showed [648]. Teachers reported that ER enhances CT and could help learning from early ages. As far as engineering is concerned, the findings were not so promising. This is not an unexpected finding considering the fact that only LEGO® Mindstorms® were involved in this study, which use readymade building blocks that students can assemble.

The third research question asked about the teachers' attitudes, concerning ER in the compulsory curricula of Greek education. As it was expected, the majority of the participants consider necessary the robotics in education. Although, there are different opinions about the age of the children that should be involved with robotics, the need for technologically improvement it seems to be recognized both by teachers and by society. However, our study suggests that it may be feasible to integrate robotics in elementary education. As it was observed, discipline was not an issue. Actually, primary school teachers participated equally other instructors.

Additionally, half of the teachers that participated in our study, believe that ER interests more the boys than girls. This is a very interesting finding reflecting the structures of a traditional society that differentiates between gender appropriate activities. Nevertheless, it seems that teachers view robots as an opportunity to engage all the students and increase interest in technology. They see a potential in educational robotics that should be explored further. However, the increased cost of robots in a setting with significant financial difficulties might discourage people from using them.

Finally, it is interesting to note teacher's T13 opinion, that any possible learning benefits will be lost if there is no follow up of the activities and further continuation, as well as connection to real industries. This is a very important comment, since there seems to be a need for altering the existing curriculum and possibly change our pedagogical practices (connection to real life activities). This is definitely something missing in the current Greek educational system where learning activities seem to be disconnected from the outside world. ER could provide a good solution to bridge this gap.

8.7 Conclusions

The present work is an observational study recording the teachers' opinions about ER from the WRO™ Hellas robotics competition. It seems that robotic activities provide the necessary means for increasing intrinsic motivation. Teachers consider that students develop numerous skills from their involvement in such activities. There seem to be numerous learning benefits for students, since apart from gaining programming and engineering knowledge, they also seem to increase their collaboration, problems solving and creativity skills as well.

Our findings imply that students in Greek education are ready for alternative teaching methods in CS and robotics might be a good addition to the present curriculum. The fact that most of them mentioned novelty, challenge and collaboration can be considered as a requirement for curriculum designers.

Although the sample of this study was relatively small, the results were quite homogeneous showing a very high level of engagement and motivation of teachers and students. The sample used in our study is biased, not only because it is limited, but also because it seems that the participating teachers were very dedicated to the robotic activities. Nevertheless, it is important to see what they mention about ER and their future in Greek schools.

CS is a large portion of the STEM field, though that one should really look at the broader topic of robotics inclusion in schools. Previous research shows that robotics is an approach to teach better our students [308, 309], but this study can be viewed as a snapshot from the limited use of ER in Greek schools. It also differs since the findings for the students are presented through the teachers' eyes.

Our future work will focus on different issues of other alternative methods like ER in CS teacher education and also attempt to record attitudes of different stakeholders like policy makers and society. We also aim to examine the impacts of the PanHellenic ER competition among a broader group of participants. As education moves forward, it is important to document teachers' motivations and experiences and to build a broader base of evidence for their perceptions about students' learning. To that end, this study builds on a wider longitudinal effort to better control questions about motivation and learning in CSE.

8.8 Elements in the ATMF

This section presents the elements of the ATMF that were affected by this research study (RS4). Figure 77 shows that the teaching method that was used is Educational Robotics.

ATMs		
Element 2		Teaching
	Section 1	Peer Learning
	Section 2	Problem-Based Learning
	Section 3	Project-Based Learning
	Section 4	Studio-Based Learning
	Section 5	Inquiry-Based Learning
	Section 6	POGIL
	Section 7	Team Learning
	Section 8	Game-Based Learning
	Section 9	Educational Robotics
	Section 10	Non-Textual Programming
	Section 11	Contextualized Learning
	Section 12	CS Unplugged
	Section 13	Subgoal Learning
	Section 14	Programming Puzzles
	Section 15	Extreme Programming
	Section 16	Program Visualization
	Section 17	Competency-Based Learning
	Section 18	Social Networks Learning
	Section 19	Emerging Technologies

Figure 77. ATM used in the RS4.

Figure 78 displays the elements of each component that were used in this study. The RS4 held at non-formal educational contexts (element 3, component 1a, dimension 1) and the supporting resources (element 2, component 1b, dimension 1) seemed to be an important factor for ER.

From the second dimension, two learner related (component 2a) elements were highlighted, gender (2) and attitudes (7). The study revealed some differences from students' attitudes about the use of robotics in education. As for the instructor related characteristics, most of the elements in component 2b were mentioned. The RS4 was about teachers' views characteristics like age, gender, teaching experience and discipline were reported.

Dimension 1: Context		
Component 1a		Education Level
	Element 1	Formal Education / Type of School / Level
	Element 2	Informal Education
	Element 3	Non-formal Education
Component 1b		Environment (physical)
	Element 1	Infrastructure (availability)
	Element 2	Supporting Resources
Component 1c		Educational system
	Element 1	Policies
	Element 2	Compulsory Curriculum
	Element 3	CS standards
	Element 4	General public attitudes
	Element 5	Time restrictions

Dimension 3: Content		
Component 3a		Learning
	Element 1	Subject
	Element 2	Objectives
	Element 3	Short-term & Long-term Planning
	Element 4	Expectations for learning
	Element 5	Relations to other subjects / disciplines
Component 3b		Teaching
	Element 1	Lecture-based
	Element 2	ATMs (see Figure 33)

Dimension 2: Participants' Characteristics		
Component 2a		Learner related
	Element 1	Age
	Element 2	Gender
	Element 3	Cultural background
	Element 4	Socioeconomic status
	Element 5	ICT use
	Element 6	ICT experience
	Element 7	Attitudes
	Element 8	Personality traits
Component 2b		Instructor related
	Element 1	Age
	Element 2	Gender
	Element 3	Teaching experience
	Element 4	Academic characteristics
	Element 5	Skills - Abilities
	Element 6	Attitudes
	Element 7	Personality Traits

Dimension 4: Evaluation		
Component 4a		Learner related
	Element 1	Knowledge / Performance
	Element 2	Motivation
	Element 3	Skills
	Element 4	Metacognition
	Element 5	Beliefs
Component 4b		Instructor related
	Element 1	Assessment Criteria / Method
	Element 2	Monitoring of Learning (Learning Analytics)
	Element 3	Feedback to Learners
	Element 4	Long-term Evaluation
	Element 5	Teaching Evaluation

Figure 78. ATMF Elements of RS4.

Moreover, from the third dimension, the element of "Relations to other subjects/disciplines" (5) from component 3a is highlighted. ER is a basic part of STEM education where many different disciplines are involved.

Finally, from the fourth dimension, three elements from component 4a were highlighted. Teachers reported on beliefs of their students about ER and the knowledge that their students acquired and the motivation that robotics gave them to learn.

8.9 Summary

In the teachers' opinions towards ER, regarding their students' involvement with robotics, during the preparation for a competition were reported. The study investigated the benefits of students' involvement with robotics about skills, motivation and learning through the teachers' eyes (qualitative data were gathered). Additionally, this chapter researched weather ER should be introduced in the compulsory curricula. The results showed that there are numerous benefits for students: they increased their collaboration, problem solving and creativity skills; understand STEM concepts about CS and engineering and especially gaining programming knowledge. In addition, most of the teachers considered that ER should be part of the compulsory curriculum.

Chapter 9

> *That is the way to learn the most, that when you are doing something with such enjoyment that you don't notice that the time passes*
> *Albert Einstein*

Research Study 5: Personal Learning Characteristics

Is there any relationship between students' cognitive style and the ability to learn programming through serious games? The aim of this research study[52] is to assess the learning effectiveness and motivational appeal of digital games for learning basic programming concepts, involving secondary education students. For this purpose, the Code.org's activity named K-8 Intro to Computer Science was used. The study (RS5) investigated students' attitudes from gaming activities to reveal the quality of their learning experience. Next, students' attitudes from games were correlated with their cognitive profile to reveal potential differences. Finally, students' performance from the digital games was assessed to reveal GBL effectiveness compared to their cognitive styles. In the study, 77 students of two Greek high schools participated in the context of the European Code Week. The results suggest that these specific games, or similar educational computer games, can be exploited as effective and motivational learning environments within schools, as they provide a high-quality learning experience. Cognitive style was found to be a significant learning characteristic that should be taken into consideration when using digital games to learn programming.

The rest of this study is organized as follows: Section 9.1.1 outlines the literature review that leads to the research hypotheses of our research. Section 9.1.2 presents the gaming activities we used for our experiment describing relevant technological and pedagogical concepts. Sections 9.2 to 9.6 present the method of the studies employed in the present article and also empirical findings. The next section (9.7) discusses the results from the experiment comparing students' cognitive profiles with games' achievement and other relations and motives regarding GBL. Section 9.8 concludes this paper, suggests the implications and limitations of the experiment, and makes recommendations for future research. Finally, section 9.9 presents the contribution of this study to the ATMF.

9.1 Introduction

In previous studies, students' attitudes (e.g., perceptions, beliefs, intentions) as far as games and learning are concerned have been thoroughly explored [270, 649]. However, the ways in which learners' attitudes are connected with some personal characteristics (cognitive style in our case) and how they affect their learning performance has not been well addressed yet. For example, Giannakos [650] identified how attitudes and learners acceptance are connected with the actual performance of the students. However, the question whether there is a relationship between students' cognitive styles and the ability to learn programming through serious games, remains. It has previously been discovered that there are significant relationships between cognitive preferences and programming related skills [651, 652]. Those findings show the significance to examine personality characteristics in

[52] Parts of this study have been published in: Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2016). How do different cognitive styles affect learning programming? Insights from a Game Based Approach in Greek Schools. *Transactions in Computing Education*. ACM. 17, 1, Article 3, 25 pages. DOI: http://dx.doi.org/10.1145/2940330.

regards to skills in various programming tasks, even when it is about computer games like in the present study. In addition, there seems to be a lack of studies combining GBL and the use of games for learning programming with the cognitive style of young students within different cultural backgrounds. To the authors best knowledge this is the first study that records the cognitive style of secondary school students and compares it to their programming skills in Greece.

Therefore, the purpose of this empirical study is twofold: (1) to examine students' attitudes on GBL and serious games that promote computational thinking and (2) to examine whether and how certain learning characteristics like cognitive style affect learners' attitudes on GBL and their gaming performance.

Our research focuses on computer games that promote algorithmic thinking and basic programming skills using games from the Code.org® website. More specifically, it describes observational data from Greek secondary education schools, using the games from the "K-8 Intro to Computer Science" activity that was designed for the "Hour of Code" (Hour of Code is organized by Code.org®[53]). The students played the games and their performance was evaluated to find possible correlations with their cognitive styles. Additionally, the different cognitive profiles were compared to students' perceptions of GBL and Computer Science (CS).

9.1.1 Background work and research hypotheses

This section presents the conceptual framework of our study, followed by the relevant literature on GBL and the quality of programming learning as well as past research on GBL, programming and personal learning characteristics.

The conceptual model of the study is represented in Figure 79. After playing the games, the student attitudes were recorded. The students reported their satisfaction levels, the games' ease of use, their perceived learning outcomes, their willingness to repeat the activity and the motivational levels. Measuring these attitudes allowed for the computation of students' Quality of Learning Experience. Student attitudes were also correlated with their cognitive styles, as it was individually assessed. Finally, cognitive style was further correlated with the students' gaming performance.

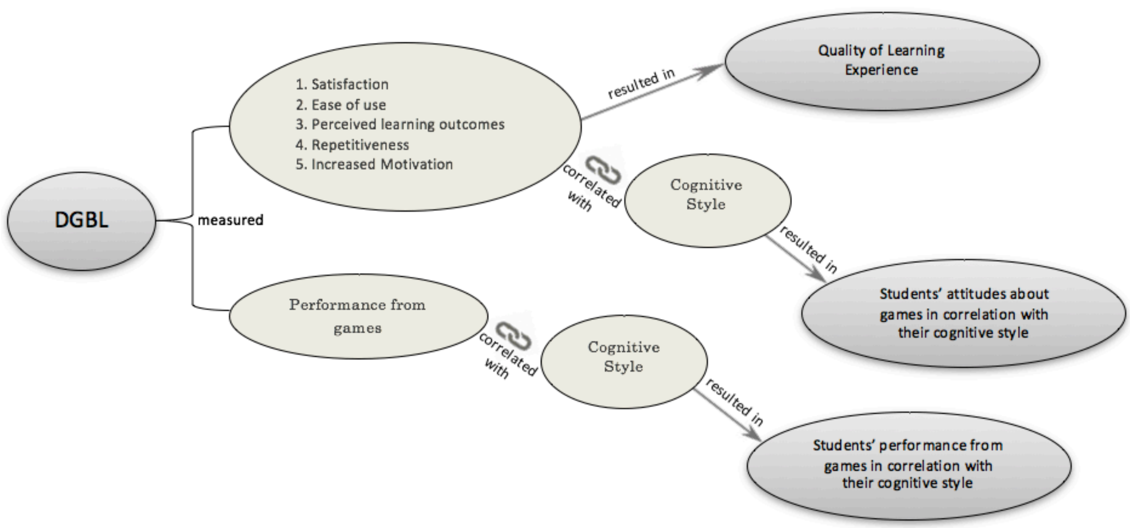


Figure 79. Conceptual Model of the RS5.

[53] Hour of Code website: http://hourofcode.com, last accessed February 2017

9.1.1.1 GBL and the quality of programming learning

With regards to Computer Science Education (CSE), games seem to be used in two main ways. There are games that students can play in order to learn how to program (GBL) and games that need to be programmed by students in order to achieve computational thinking (Game Design). In the former case, games are preexistent to the student's activity whereas in the latter, games are the result of the student activity. Both approaches increase computational thinking and information technology fluency [653].

For games that need to be programmed by students there seem to be formal evaluation methods to assess students' learning outcomes, like the Game Computational Sophistication model [654]. Previous studies have shown very positive results for Game Design in CSE [653, 654] and in particular student computational thinking skills and motivation increased significantly [655]. These positive study results lead to the creation of a scalable game design curriculum in order to provide a holistic framework for CSE in the American schools [655]. Moreover, it was found that when students design games to learn programming they have clear educational benefits, like motivational increases [656]. Similarly, when games were used to teach programming skills (GBL), the results were also very positive [270]. Thus, in both cases where games are either created by students or used by students for programming, the results are very positive for both genders [270]. In the present study, a GBL approach is used, where games are already developed and available to be used by students.

In addition, there seems to be many different approaches in the teaching of CSE in different countries. Previous studies have investigated the role of the cultural background in programming activities [657], finding significant cultural differences. For this reason, different cultural approaches are collected and compared under the scope of computer science education [658].

For all the above reasons, it was decided to observe GBL and report the results of Greek Secondary Education school students using games in order to learn basic programming concepts. Therefore, we hypothesize that:

H1- Students' learning experience through the "K-8 Intro to CS" gaming activities will provide them with a high quality experience, on the secondary educational level.

Observational data are reported under H1. The quality of an experience in general, provides an assessment of human expectations, feelings, perceptions, cognition and satisfaction with respect to a particular product, service or application [659, 660]. Therefore, Quality of Learning Experience (QoLE) can be represented by several and various variables. As shown at the study's conceptual model (Figure 1), we worked with five attitudinal categories (Satisfaction (SAT), Ease of Use (EOU), Perceived Learning Outcomes (LEA), Willingness to Repeat GBL (REP) and Increased GBL Motivation (MOT)), in order to reveal the QoLE. In the present study, important and widely used axes were employed to build a post-activity questionnaire and approximate the students experience through games (Table 40).

Satisfaction measurements have been used by previous studies to evaluate the learning potential and outcome in CSE [661]. In addition, it seems that students appreciate alternative teaching methods and report increased satisfaction and encouragement [662]. Moreover, ease of use and in general usability, is commonly checked in CSE through games, either GBL and/or Game Design. For example, students reported the usability of games they designed [663]. Additionally, the Technology

Acceptance Model (TAM) has been a major tool for studying how users come to accept and use a technology [664]. With respect to the TAM, perceived usefulness and perceived ease of use have been shown to be factors in explaining user acceptance behavior towards using a system. It was therefore important to assess the games' ease of use.

Table 40. Quality of learning experience for this study and explanation

QoLE Axes	Description in our questionnaire
Satisfaction	*I enjoyed the "Hour of Code" gaming activities*
Ease of Use	*The "Hour of Code" gaming activities were easy*
Perceived Learning Outcomes	*I acquired some basic programming principles by playing the games*
Willingness to Repeat GBL	*I want to repeat similar gaming activities in the future*
Increased GBL Motivation	*My learning motivation is bigger when I play digital games*

Furthermore, in CSE one way of measuring the learning outcomes is by recording students' perceptions of their own learning. Students' perceived learning outcomes have been used in the past to assess the QoLE of the students [665].

Looking at the use of serious games through TAM related theories [666] we can assume that the practical use of a digital game system is determined by the users' intentions to use the game (to play). In our case, the use of games is determined by the users' attitude toward repeating learning through playing.

Finally, an important aspect of learning activities is to increase the motivational levels of students [667]. Games in particular are known to increase learning motivation [668].

9.1.1.2 GBL, programming and personal learning characteristics

Aiming at GBL, it was significant to study issues relevant to human learning and especially related to learning with the use of technology. Post-modern theorists like Hlynka [456] believed that educational technology is not neutral in the sense that it does not simply support a learning process by making it more efficient but it changes the very nature of learning. Leblank et al. [669] showed that the use of educational technologies indeed changes the nature of learning tasks. There is a number of factors that can significantly influence an individual's learning. Many theories have evolved around this issue and a short description is provided here. We have divided the different factors in two main categories; 1) factors that affect learning and are situation independent and 2) factors that affect learning and are situation dependent.

In the category of situation independent learning characteristics, we primarily place cognitive style and intelligence type. These two factors have a strong relation to the individual's personality and they remain relatively constant over situation and time, or at least they are not that easily influenced by the different learning situations. Cognitive style is a person's preference and habitual approach to the organization and representation of information [118]. Different researchers have described different aspects of cognitive style [118, 119, 605-608, 670]. Cognitive style is a research construct assisting the study of cognitive issues related to learning. Students whose learning methods and material matches their cognitive style perform better [671]. Another situation independent learning factor is the intelligence type [672]. Other situation independent factors are: age, pre-knowledge, maturity, confidence, preferences, background, responsibility, gender, attitudes and working memory [673]. From all situation dependent factors, cognitive style is the one most used in technology related educational matters and the one we will focus in the present research.

Similarly, some known situation dependent learning factors are: approach to learning [674], motivation type [675] and learning style. We focus on learning style since it is a well-studied and widely used situation dependent learning factor. Learning style has a situational component as well as an internal, cognitive aspect. Thus, it can change from time to time and from task to task.

"Each individual responds differently to a learning situation. This response will be influenced by the way the individual thinks, her past experience, the demands of the environment and the current tasks. This approach is generally recognized as the individual's learning style" [676].

Perhaps the most known and influential theory on learning style is Kolb's experiential learning theory [110]. Learning style thus is easily influenced by student external factors, such as instructor's personality, perceived difficulty of subject matter, etc. For the purposes of the present study, cognitive style was chosen for three main reasons:

1) It is a relatively constant characteristic that remains stable across different learning situations, since it describes deeper cognitive structure than learning style. This means that each learner can be assessed for her cognitive style once and this information can be applied across different subjects and learning situations.

2) It can be easily assessed with a use of a short questionnaire (we used a short version of 28 questions). In the present work, the assessment tool used was MBTI (Myers-Briggs Type Indicator) [677]. The MBTI is based on Jung's theory of psychological types and it describes learners on four dimensions based on self-reported questionnaires. The MBTI is mainly used by organizations for employee selection and placement and it has a strong validity and reliability.

3) According to past research, deeper learning structures such as cognitive style are better to use for technology related educational matters than others like learning style [676].

With regards to GBL and personal learning characteristics, previous studies have provided conflicting results. For example, Batista and Cornachione [678] studied digital game satisfaction in business games and found no significant differences in terms of individual learning styles. However, Sung et al. [679] found that the learning motivation of students, who learnt with a learning version that fits their learning style, was significantly higher. In addition, when researchers studied programming performance in GBL the results also indicated that personal learning characteristics are indeed important. For instance, Similarly, Lau and Yuen [680] found that students of a certain learning style (sequential learners) performed better than students of a different learning style (random learners) in programming tasks. Finally, Milovanović et al. [681] recorded the cognitive style of university students who engaged in game design for understanding computer networks. The study showed that GBL was very effective and that this effectiveness depends on specific cognitive styles.

From the above, there seems to be a lack of studies combining GBL and the use of games for programming (not game design) with the cognitive style of young students. To the authors best knowledge this is the first study that records the cognitive style of secondary school students and compares it to their programming skills in Greece. Thus, we will study perceived quality of learning experience in the light of cognitive style and will compare the game learning outcomes to the cognitive style of students. Consequently, we hypothesize that:

H2 Students of different cognitive style have different attitudes concerning the quality of their learning experience with GBL and programming.

H3 There are significant differences in the students' performance depending on the cognitive style.

9.1.2 The game activities

For the purpose of this study, Code.org®'s "K-8 Intro to CS" gaming activities were used. Code.org® hosts numerous serious games, since their purpose is not purely entertaining but mainly educational [624]. "K-8 Intro to CS" activity, is amongst the most popular and it is a free online course. It aims at showing students that programming can be fun and creative[54]. As stated by the game's designers[55], the course is designed to motivate students and educators to continue learning computer science to improve real world relationships, connections, and professional life.

"K-8 Intro to CS" activity is proposed to be a 20-hours course and introduces core computer science and programming concepts. Although the course is designed for use in classrooms for grades K-8 (that means from kindergarten up to 8th grade, age 14, which at the Greek educational system corresponds to junior high school), it can be used by all age groups. The game activities do not require specific prerequisites in curricular school subjects or any prior programming skills.

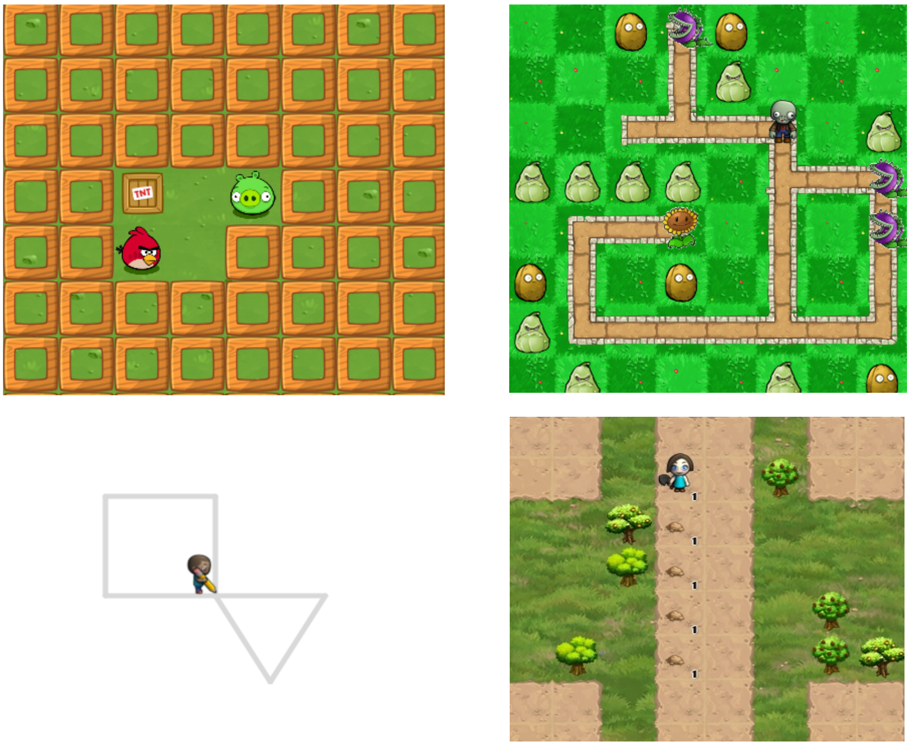


Figure 80. Screenshots from Code.org® games.

The game consists of numerous creative activities (puzzles) that are designed to teach computational thinking and the basics of computer science concepts. The game also enhances problem-solving

[54] Code.org®'s - Teach our K-8 Intro to Computer Science: http://code.org/educate/20hr, last accessed December 2016

[55] Code.org®'s proposed school curriculum: https://code.org/educate/curriculum and https://code.org/educate/curriculum-philosophy, last accessed December 2016

abilities. The course was developed in accordance with Code.org®'s educational philosophy and Computer Science Teacher Association's standards (Common Core, NGSS and CSTA standards[56]).

In each puzzle, students "wrote" a program that got a character through a maze or created some geometrical shapes. The characters were from the popular kids' games "*Angry Birds*" and "*Plants vs. Zombies*"[57] (Figure 80).

All the puzzles are based on visual programming languages (VPL) [682, 683]. In general, there are two primary groups of programming languages: text-based programming languages and VPL. Languages like JavaScript or Python are text-based but Code.org®'s intro course uses Blockly, which uses visual blocks that students drag and drop to write programs. Such visual programming environments with phenomenal success are Scratch [684] and Alice [357]. Using such a visual language is easier to learn programming versus a text-based language because the students do not have to think about the syntax of writing programs but only focus on the logic of how to write programs. As Barth[58] states about Google's Blockly programming project for education, "by minimizing the use of syntax, users are able to focus on the logic and concepts used by computer scientists, progressing at their own pace as they venture through mazes and more advanced arenas". Although students used blocks to write code they were able to see the code they wrote represented in JavaScript, a text-based language. A brief explanation of the key concepts that are taught by the games and also the basic blocks that are used, stands at the appendixes section.

9.2 Method

9.2.1 Context

In order to be able to compare our research results in the field of CSE in schools of different countries and to provide a better framework for our study, the Darmstadt model was used [39]. Since a whole analysis of the Darmstadt model is outside the scope of the present work, a short description of preconditions, circumstances and influence factors in the Greek educational system are presented, to provide a basic framework for our study.

In particular, the Range of Influence (as local) is:

Country: Greece.

Region: Arcadia.

Schools: Public High Schools.

Classrooms: 1st, 2nd and 3rd classes of Junior High school and 1st class of Senior High school).

The Educational Relevant Areas axes are as follows:

Socio-cultural factors: CS courses are not considered primary in Greek secondary education both from the society and the curricula.

[56] Code.org®'s Hour of Code activity maps to CSTA Standards : http://csta.acm.org/Curriculum/sub/K12Standards.html and https://docs.google.com/document/d/16UYAETJBWLQ3e9HyEkTJjlqgyAF3bU5n6RHV2GZqv_Y/pub, last accessed December 2016

[57] Angry Birds, Bad Piggies and all related properties, titles, logos and characters are trademarks of Rovio Entertainment Ltd. Plants vs. Zombies and PopCap are trademarks of Electronic Arts Inc.

[58] Google's Google Introduces Kids to Coding Through Blockly Games Project: http://www.eweek.com/it-management/google-introduces-kids-to-coding-through-blockly-games-project.html#sthash.WFnPB8e0.dpuf, last accessed December 2016

Policies: due to the economic crisis in Greece, there is lack of CS teachers and laboratory infrastructure in many cases mostly in provisional schools.

Teacher Qualifications: CS teachers are highly qualified with University degrees. However, they come from several disciplines (e.g. physics and mathematics), due to a lack of teachers qualified purely teachers in informatics, when CS courses entered the curricula.

Curriculum Issues: CS courses are taught only one hour per week in grades seven to nine. In tenth grade (first year of senior high school in Greece), CS is an optional course. In eleventh grade, it is taught an hour per week and finally in the last year of secondary education, it is taught just in specific scientific fields.

Examination/Certification: CS courses in Greek secondary education do not lead to any certifications.

Motivation: In Greek educational system, students finishing the secondary education, take national exams in order to enter higher education. A programming course is a prerequisite to enter economic and computing universities.

Teaching methods: All CS courses are considered laboratory courses and are taught in school labs. However, they are taught with traditional teaching methods and computers are not used very often.

9.2.2 Participants

In this study, seventy-seven secondary school students (*N=77*, Age 14-18, M = 15.32, SD = 1.09) participated. Participants were recruited voluntarily, in the context of the CSE Week[59]. Students of four classes were invited to participate and all *N=77* students of the four classes volunteered. Informed permission was taken from all students and their parents. Among the respondents, 50.6% were female (*N = 38*) and 49.4% were male (*N = 39*).

The students were from two Greek high schools; the first was a junior high school (Gymnasium in Greek) and the other a senior high school (Lyceum). In Greece, there are six grades in secondary education and the students that participated in this study were from the first four grades. Both schools are located at a small county city at central Peloponnese.

Working with secondary education students, implied that there was limited access to their tight schedule. That meant that there was no option of running a pre-test analysis on their computing knowledge prior to the study, apart from self-assessments. To access the computing base knowledge of the participants a specific question was included in the questionnaire asking their assessment of their computing experience and skills. All participants had limited experience in programming as they were introduced to the Logo programming language in the 8th grade for one hour weekly during the obligatory CS lesson. Some of them had also used Scratch through the school's educational programs but only at an introductory level.

All students that enrolled at the study (*N = 77*) followed up as planned through each stage of the study and they received all intended interventions from researchers. The participant flow remained constant, since there were no dropouts between the different stages of the activities.

[59] CSEd Week program: http://csedweek.org and http://csedweek.org/about, last accessed December 2016

9.3 Study 1 (H_1): Students' attitudes towards serious games that promote programming (QoLE)

In the first study, we investigate H_1 which consisted of five axes; a positive result would conclude that students had a positive experience with the games and therefore that the games provide high QoLE.

9.3.1 Research design

In our effort to investigate the QoLE of the "*K-8 Intro to CS*" gaming activities, we conducted a study that measures students' attitudes on five factors after having played the games. The research design consists of five steps (Figure 81) and each one of the measures results in students' attitudes from games (SAT, EOU, LEA, REP, MOT).

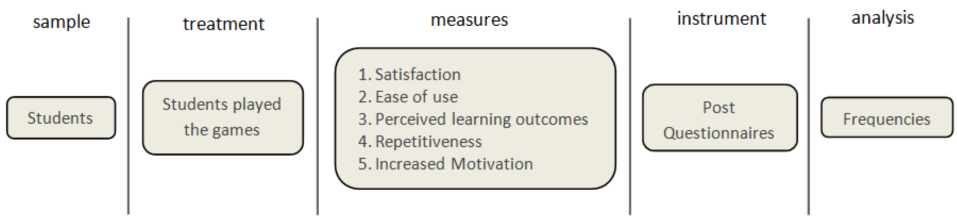


Figure 81. Graphical Representation of the research design of the 1st study.

9.3.2 Procedure

In the context of the Code Week we organized a series of workshops for the students of the two secondary schools, outside school hours. In order to complete the study, students worked for 4 weeks through a tangible and well-framed timetable that was initially presented in the course description. All the meetings were held at the schools' computer labs, where students could use the labs' computers or their personal laptops. A teacher (CS) and the researchers coordinated the students. The experiment consisted of a start-up meeting, two main gaming sessions and a wrap-up meeting where students got certificates for participating in the study.

All the students engaged with the games seven hours in total within the 4 weeks. The goal was to complete as many stages as possible by sharing the same time and resources. Thus, although they could access their accounts from anywhere and complete the puzzles-activities, it was decided that they only access their accounts from the school's laboratory. All participating students worked until they had completed the pre-agreed time.

While students were doing the activities, the teacher and the researchers walked around the classroom and answered any questions they might have. In addition, they identified students that were far behind and helped them more. Generally, the students were advised to think about the logic of the puzzle instead of just adding more blocks when they came across difficulties. They were also asked to explain in words what they wanted to do. Finally, they were encouraged to ask for help.

9.3.3 Measures and data analysis

Data for H_1 were collected from a feedback questionnaire and referred to the quality of the students' learning experience. After students' enrollment with the gaming activities, they completed a paper-based survey[60] where they had to choose between simple answers based on a five-point Likert scale in order to find:

- perceptions about CS and programming,
- perceptions about the gaming activities,
- motives regarding digital games and GBL.

Data were analyzed in using the SPSS statistical package: Frequencies were used to analyze H_1.

9.3.4 Research Findings

Using frequencies analyses, to each one of the five partial axes of H_1, the results concerning students' motivation and attitudes towards the games are presented in **Error! Reference source not found.**.

Table 41. Statistics on students' quality of learning experience.

		SAT	EOU	LEA	REP	MOT
N	Valid	77	77	77	77	77
	Missing	0	0	0	0	0
	Mean	3.91*	3.49*	3.53*	3.90*	3.66*
	Std. Deviation	.948	.927	1.059	1.033	.995

* 5-point Likert was used at the questionnaire. Top score was 5.

As shown at Figure 82 the overwhelming majority (81.8%) of the students enjoyed the gaming activities. More specifically, *N=44* students chose agree, that is 57.1% and additional *N=17* chose strongly agree namely 24.7%. Most of them also thought that the specific gaming activities were easy to use (68.8%).

Figure 82. Participants' Satisfaction and Ease of Use.

Furthermore, most students believed they acquired some basic programming principles by playing the games (65%) and also wanted to repeat similar actions in the future (72.8%) (Figure 83).

[60] Can be found online http://goo.gl/forms/OOUiMUO4Eo

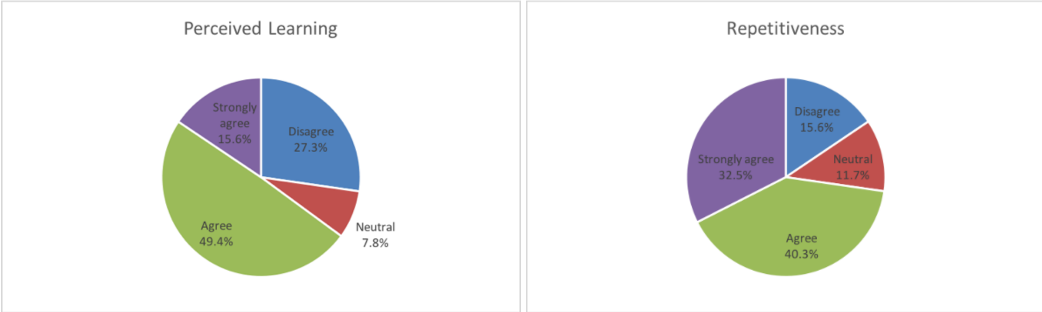


Figure 83. Participants' perceived learning and intention to participate in similar activities.

Finally, it was widely believed that students have higher motivation to learn by playing computer games (62.4%) (Figure 84).

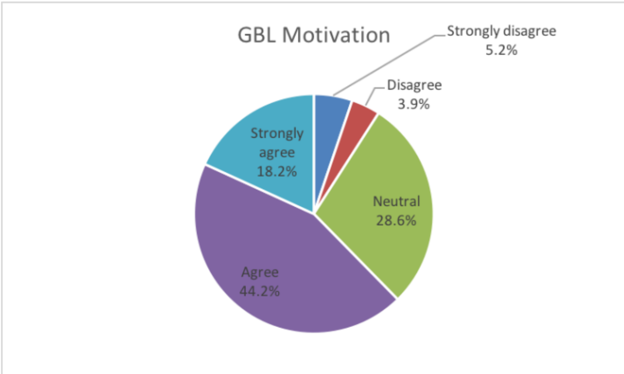


Figure 84. Participants' motivation.

Consequently, H_1 that games ("*K-8 Intro to CS*") can positively affect student learning and motivation and provide a high quality learning experience is supported.

9.4 Study 2 (H_2): Correlation of students' QoLE with their Cognitive Style

Wishing to investigate whether student perception on their Quality of Learning Experience is affected by their personal cognitive style, a further analysis was performed, as described in H2.

9.4.1 Research design

In our effort to investigate potential students' differences as long as cognitive style is concerned and their attitudes about 1) Satisfaction, 2) Ease of Use, 3) Perceived learning outcomes, 4) Willingness to repeat GBL and 5) Increased GBL motivation, after having played the games, we conducted a study that correlates students' QoLE with their personal cognitive style. The research design consists of five steps as shown at Figure 85.

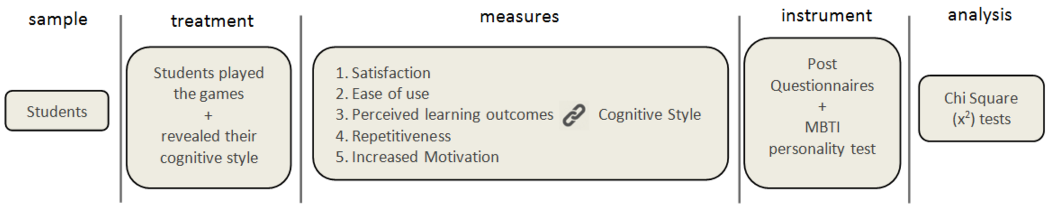


Figure 85. Graphical Representation of the research design of the 2nd study.

9.4.2 Procedure

The procedure that was followed for the students QoLE is described above (study 1). This study correlates the results from study 1 with students' cognitive profile.

For that reason, a personality test (MBTI) was used. The MBTI test provided results in 4 dimensions with two possible values each (16 possible types) in order to identify their cognitive profile (Table 42). The first dimension, **E**xtraversion (**E**) and **I**ntroversion (**I**), explores the way that students are energized. The second dimension explores type of information that students prefer to pay attention to (**S**ensing (**S**) and i**N**tuition (**N**)). The third MBTI dimension, investigates the way they prefer to make decisions (**T**hinking (**T**) - **F**eeling (**F**)) and the final dimension with the terms **J**udging (**J**) and **P**erceiving (**P**), describes how a person deals with the outer world.

Thus, students completed a paper-based test that revealed their cognitive style. As a motivator, once the cognitive style of each student was determined, information about famous people with the same style was also provided[61]. Students seemed to enjoy this process as it was depicted by their reactions.

Table 42. MBTI Dimensions.

Extraversion (E)	–	(I) Introversion
Sensing (S)	–	(N) INtuition
Thinking (T)	–	(F) Feeling
Judging (J)	–	(P) Perception

9.4.3 Measures and data analysis

For H_2 we used the data form the feedback questionnaire and data from the personality test. Chi Square (X^2) tests for independence were used to investigate whether distributions of categorical variables correlate with others. These tests were used to investigate potential initial correlations regarding students' cognitive profiles. The level of significance was set at 0.05.

The scale of the questionnaire regarding the factors (SAT, EOU, LEA, REP, MOT) that were used for H_1, had a high level of internal consistency, as determined by a Cronbach's Alpha of 0.769.

9.4.4 Research Findings - Results

In this section, the results comparing the students' cognitive profiles and the performance on computer games are presented. The relations concern H_2 and they are presented based on the four cognitive dimensions (Table 42). In addition, Pearson correlations were performed for all the MBTI dimensions. Our study clearly points out (Table 43) the significance of the MBTI cognitive dimensions.

[61] Celebrities according to psychological types: www.celebritytypes.com, last accessed February 2017

Table 43. Pearson's correlation coefficient between MBTI dimensions.

MBTI dimensions	E-I	S-N	T-F	J-P
E-I	1			
S-N	0.003*	1		
T-F	0.472	0.007*	1	
J-P	0.116	0.000*	0.065*	1

*. Correlation is significant at the 0.01 level.

The first MBTI dimension, Extraversion (E) - Introversion (I), and the third dimension (Thinking (T) - Feeling (F)) did not provide any significant results at this study. In the second dimension Sensing-iNtuition there was a significant relationship between students' cognitive style preference (S-N) and the perspective of easiness that students had towards the "*K-8 Intro to CS*" game activities they used, $x2$ (3, $N=77$) = 11.788, $p=.0008$ (Figure 86). Students with high intuition found the gaming activities easier.

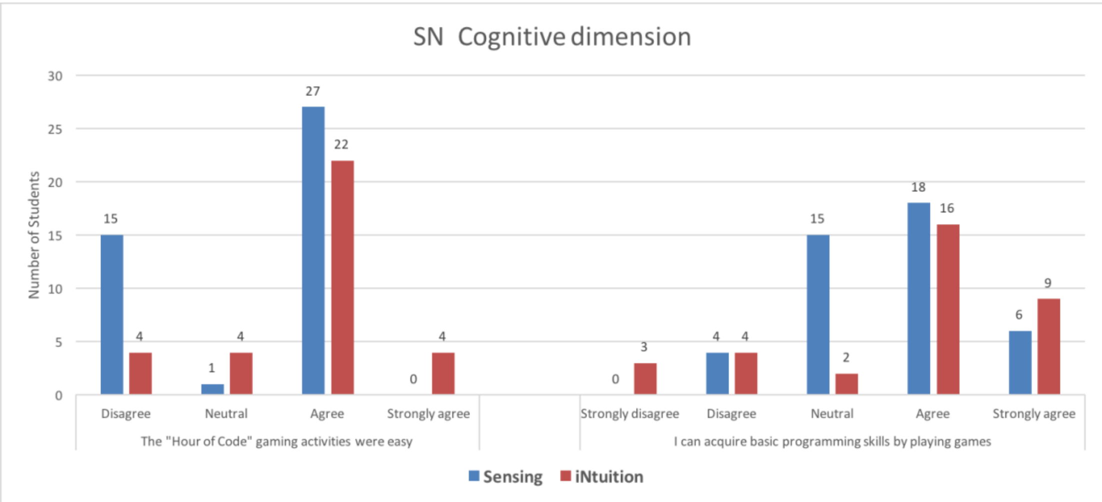


Figure 86. Crosstab - S-N cognitive dimension with EOU and LEA.

Moreover, about the preferences S-N there was a significant correlation between students' perceptions about acquiring basic programming skills from GBL, $x2$ (4, $N=77$) = 12.781, $p=.012$ (Figure 86). Sensing students consider more that they can learn basic programming skills by playing computer games like those of the "*K-8 Intro to CS*".

Finally, as far as S-N dimension is concerned, there was a significant relationship between the preferences Sensing-iNtuition and students' attitude towards motivation in GBL, $x2$ (4, $N=77$) = 13.437, $p=.009$ (Figure 87). Students with high intuition found the gaming activities more motivating.

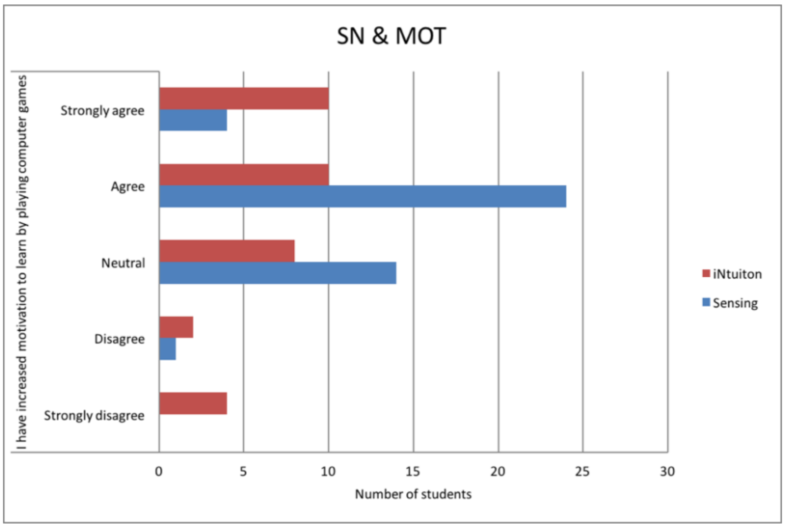


Figure 87. Crosstab - S-N cognitive dimension with GBL motivation.

As for the fourth dimension, Judging-Perceiving, there was a significant correlation between students' perceptions about acquiring basic programming skills from GBL, $x2(4, N=77) = 21.384$, $p=.000$ (Figure 88). Judging students consider that they can learn basic programming skills by playing computer games.

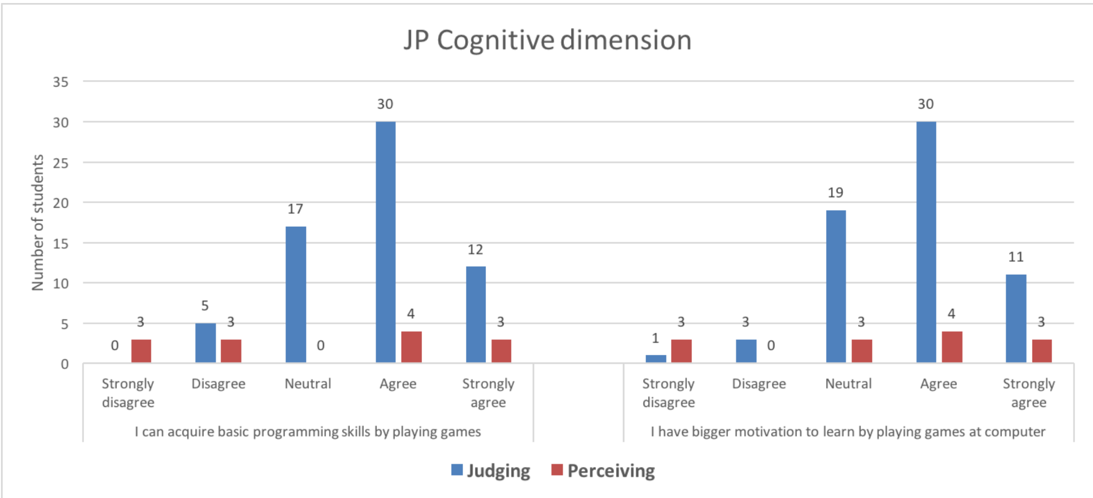


Figure 88. Crosstab - J-P cognitive dimension with LEA and MOT.

For the same cognitive dimension (J-P), a significant relationship between the students' perspective on learning effectiveness of basic programming skills and playing computer games was found, $x2(4, N=77) = 11.243$, $p=.024$ (Figure 88). Judging students seem to have higher motivation to learn when playing computer games.

9.5 Study 3 (H_3): Correlation of students' performance from games with their Cognitive Style

Wishing to investigate a possible relationship between students' cognitive style with their gaming performance, further correlational studies were performed, in order to study research H_3.

9.5.1 Research design

In our effort to investigate potential students' differences on their performance from the gaming activities as long as cognitive style is concerned, we conducted a study that correlates students' assessment from GBL with their personal cognitive style. The research design consists of five steps as shown in Figure 89.

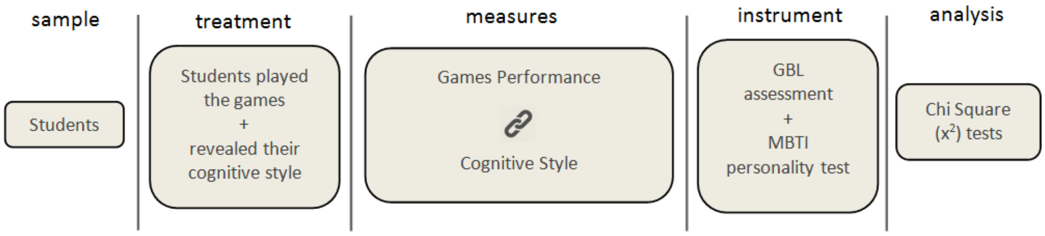


Figure 89. Graphical Representation of the research design of the 3rd study.

9.5.2 Procedure

Initially, we created a class at the website of code studio[62] where students had their account, so that everybody could log in later and continue the games. By doing so, we could administrate the whole process and monitor students' progress anytime through the student's dashboard. The dashboard displayed the lessons that were available, and gave us feedback about their progress through the lessons.

The score from the digital activities was calculated with a procedure based on the official marking provided by the Code.org®'s dashboard (by the time the activities were completed December 2013 - Hour of Code 2013). By clicking on the "*Progress Dashboard*", teachers could check students' progress in a very detailed manner included progress in the last hour. Figure 90 shows a screenshot from the teacher's view dashboard.

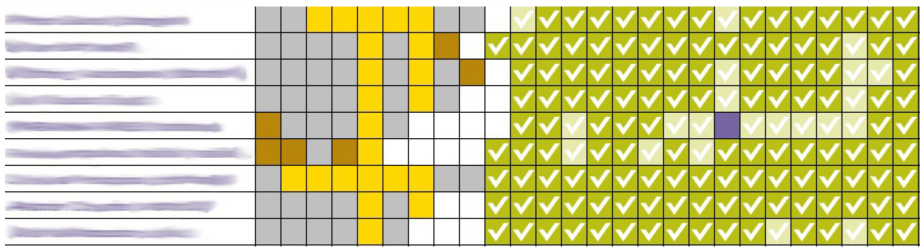


Figure 90. Partial view of teacher's dashboard (Hour of Code 2013).

A mouse over action revealed the points gained, as shown in Table 44. A dark green square means that the student completed the exercise correctly and used the blocks required. A light green square means that the student used too many blocks of code to complete the exercise and may do it again. In that case, the marks she achieves are lesser than the previous case. Each stage reports the number of ideal blocks that the students should use.

Table 44. Calculation for students' performance from games.

Possible conditions for each stage	Marks	Comments	Symbol at dashboard
Completed, perfect	100	for best result	green - white check
Completed, but using too many blocks	20	for best result	light green - white check
Tried, but not completed	4	regardless of attempts	blue square
Not tried yet	0	--	white square

[62] Accelerated course (K-8 Intro to CS): http://studio.code.org/, last accessed February 2017

Table 45 shows the nine different activities; each stage consisted of several stages (e.g. the "*Maze*" activity consists of 20 stages). Supposing that a student finishes all stages of all the activities with perfect marks, the score she achieves is 9800.

Table 45. GBL assessment through the puzzle gaming activities.

"K8 Intro to CS" gaming activities	Stages	Max Score
Maze	20	2000
Artist 1	10	1000
Artist 2	11	1100
Farmer 1	11	1100
Artist 3	11	1100
Farmer 2	10	1000
Artist 4	10	1000
Farmer 3	9	900
Artist 5	6	600

Then the marking gap between the perfect and the lowest was divided in four equal sets. That allowed the student scores to be handled easier since there would be four main categories of scores (bad, good, very good and excellent) (Table 46).

Table 46. Simplified students' overall progress from gaming activities.

Value	GBL Score	Characterization
1	0 – 2450	*bad*
2	2451 – 4900	*good*
3	4901 – 7350	*very good*
4	7351 – 9800	*excellent*

The grading scale was also similar to the Greek schooling system, where different grade ranges follow specific descriptions, "*Bad*", "*Fair*", "*Good*", "*Very Good*" and "*Excellent*". Therefore, the scores corresponded to the nominal scale (1=bad to 4=excellent) and not the numerical (1-9800) and by that way were also more understandable to the students. Finally, apart from the ease of handling the student scores, another reason for this practice was the fact that some students failed to complete certain stages. Instead of assigning a grade of zero only in these stages, the way the score categorized (Table 46), produced a grade better tuned to the students' performance. Therefore, the four categories associated in such a way to differentiate the sample of the research without altering the statistical validity.

9.5.3 Measures and data analysis

Data were analyzed using the SPSS statistical package: Chi Square (x^2) tests for independence were used to investigate whether distributions of categorical variables correlate with others. These tests were used to investigate potential initial correlations regarding students' performance from games and their cognitive style. The level of significance was set at 0.05.

Additional, students who participated in the study were from different age groups, implying that our sample might not be homogeneous. Therefore, we run a chi-square analysis in SPSS, in order to find possible correlations on students' performance form games with ages, and we found that their achievement at the gaming activities was not affected by the age factor. The age filter showed no significance results.

9.5.4 Research Findings - Results

A significant relationship between the preferences Extraversion-Introversion and the overall progress that students achieved by playing the computer games was found, $x^2(3, N=77) = 14.427, p=.002$ (Figure 91). Introvert students performed better than Extraverts.

Additionally, there was a significant relationship between the preferences Judging-Perception and the overall progress that students achieved by playing the computer games, $x^2(3, N=77) = 10.307$, $p=.016$ (Figure 91). Judging students were also connected with higher performances.

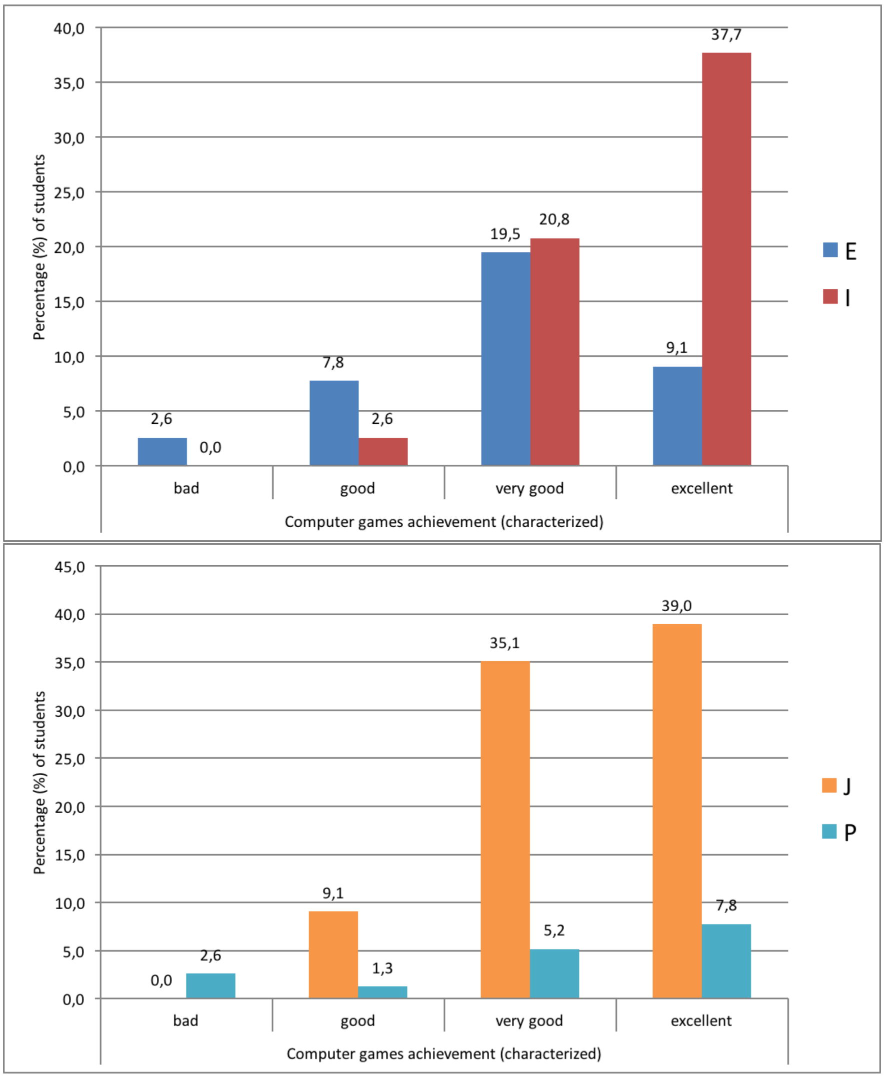


Figure 91. Crosstab – Cognitive style with Games achievement.

9.6 Additional findings

There were also various significant relationships that although not directly related to the initial hypotheses, nevertheless they provide important information regarding cognitive style and learning aspects (Table 47). Those findings come from some extra questions at the post-survey that students answered after the gaming activities.

Table 47. Chi square analyses with further findings regarding cognitive styles.

MBTI Dimensions	Correlation	Value Asymp. Sig.	Conclusion
Extraversion Introversion	School Performance	$x^2(2, N=77) = 13.866$, $p=0.01$	Introvert students seem to achieve higher performances
	Use of Computers	$x^2(5, N=77) = 13.171$, $p=0.02$	Introvert students use computers more often
	Learn more about CS	$x^2(4, N=77) = 9.786$, $p=0.04$	Introvert students want to learn more about CS
Sensing iNtuition	Use of Computers	$x^2(5, N=77) = 11.531$, $p=0.04$	Sensing students seem to use computers more often
	Playing Computer Games	$x^2(4, N=77) = 13.062$, $p=0.01$	Sensing students seem to spend more time on playing computer games
Thinking Feeling	Playing Computer Games	$x^2(4, N=77) = 9.627$, $p=0.04$	Thinkers seem to spend more time on playing computer games
Judging Perception	Playing Computer Games	$x^2(4, N=77) = 10.168$, $p=0.03$	Judgers spend more time on playing computer games

9.7 Discussion

This section provides both theoretical and practical explanations of the observed results. Our study examined the cognitive mechanisms that underlie learning and thinking in the settings of CS with the use of computer serious games. The concluding aim of the described experiment was to investigate whether: 1) digital games provide a high quality learning experience to students (H_1), 2) students' cognitive style is a factor that differentiates the way they perceived the learning experience through games (H_2) and 3) some students achieve better result through gaming than others as far as cognitive style is concerned (H_3).

We found that H_1 was confirmed and this enhances the perception that GBL can be used to provide high quality of learning experience. More specifically the majority of the students enjoyed the gaming activities. Although this was expected from the literature, students enjoyed the games and participated happily in this study even outside school hours. Most students found the games of "*The Hour of Code*" easy. As mentioned above the "*K-8 Intro to CS*" gaming activities that we used at this study are addressed mainly for younger children (in this study all the participants were from secondary education). Thus, the students' perception about the difficulty justifies this finding. Moreover, the results here show the usability of Code.org®'s games. It is also very important, that most students believed that they acquired some basic programming skills after having played the games. Generally, students considered that programming can be taught through GBL. Finally, the majority of the participants wanted to repeat similar activities in the future and also were very motivated, and engaged in the GBL tasks. It seems that students are ready to incorporate alternative learning methods together with the traditional ones.

The H_2 and H_3 hypotheses, were both confirmed, since the cognitive style of the students seemed to affect their learning attitudes and their gaming performance. More specifically, Introverted students achieved higher performances in the GBL. Although there are literature findings showing the introversion is correlated with better school performances [685] and also found here, to the best of the writers knowledge no such finding has been previously found in educational games. Students that tend to focus at the inner world of ideas and impressions (i.e. introverts) do better with educational games. It is useful at this point to mention that Introverted students of this study use the computer more often than Extraverts, as shown at the 4.6 section about additional findings of the study.

As far as the S-N dimension is concerned, students with high iNtuition found the gaming activities easier than Sensing students. Usually, people with high intuition tend to focus on patterns when they gather information [686, 687], which seems to explain this finding. In other words, programming games are based on patterns and according to cognitive style expectations, iNtuition students were expected to perform better, as they did. Although Sensing students in this study were spending significantly more time using the computers at home than iNtuition students, it was students with high intuition that found the activities easier. It seems here that only the duration of exposure to the medium is not adequate to increase learning efficiency and thus, learning activities need to match certain cognitive structures as well. In addition, Sensors consider that they can learn basic programming skills by playing computer games, more than those with high intuition. Generally, those who prefer sensing tend to focus on the present and on concrete information gained from their senses [686, 687]. Expanding this argument, it seemed that the way the students gathered the learning information serves better the Sensing ones.

Finally, for the J-P dimension, Judger students also were associated with higher performances than the Perceivers. In addition, students who prefer judging consider that they can learn basic programming skills by playing computer games and it seems that they have higher motivation to learn by playing computer games. It appears that a planned and organized approach, like the tasks used in this study, suits better the Judgers. On the other side, students with higher perceiving preferences tend to like a flexible and spontaneous approach of learning than a single technique like GBL [686, 687]. Lastly, the Judgers' were found to spend significantly more time playing computer games and that might enhances the finding that they achieved higher performances from games.

9.8 Conclusions

In this research study, we investigated the relation of GBL to promoting programming in secondary education students. The study was organized in three phases. First, we found that the digital gaming activities ("K8 Intro to CS") provided a high quality experience to most students. In the second phase, we revealed students' cognitive style and correlated it with their attitudes from the first phase. The findings of this phase indicate that students' personal learning characteristics as cognitive style differentiated the way they perceived learning through the games. Finally, in the third phase we assessed students' performance from the games and correlated it with their cognitive styles. Students of certain cognitive styles seem to outperform others of different cognitive styles in regards to game-based programming.

9.8.1 Guidelines for curricula, game designers and teachers

CT is an important 21st century skill. CS can also provide opportunities for career development and in that light it should be incorporated in contemporary curricula. Code.org®'s "*K-8 Intro to CS*" or commonly called "*Hour of Code*" seems that it can provide high quality learning experience to teach kids programming. Therefore, the way to integrate it into formal education settings should be further investigated. The potential power of such digital games in the classroom is becoming more obvious. Such gaming approaches could be used as an additional teaching and learning tool together with other traditional schooling activities. Our study incorporated the games at secondary education level, but games could be used with younger children too.

However, the particular games used in the present study seem to suit better specific cognitive styles and in particular Introverts and Judgers. Although most educational games mainly focus on the pedagogical aspects, game design issues regarding different types of users, are often neglected. The criteria for design and evaluation of the games needs to be expanded. The need to balance these dimensions is identified in the literature [688]. Nevertheless, it seems that there are technologies focusing on learning goals, teaching styles, learning approaches, instructional guidelines [689, 690], but none focuses on cognitive style, thus in the deeper personality structures of the learner.

CS teachers should instruct students with different styles in a more personalized and efficient manner. Possible, a well-organized curriculum and some developments at the games' design could accommodate more cognitive styles. Moreover, if games engage more students in the learning process then students can learn from playing games. Towards the "learning with the game" direction, some scenarios-suggestions that would be suitable for most cognitive styles can be:

1. *Feedback from Games*: The games we used, provided feedback based on student performance, which in general support Thinkers. However, feedback should be also provided based on effort invested, to support Feelers too. In that light, the games could provide positive feedback for perseverance, meaning that a student get positives scores even if she fails, as long as she keeps on trying.

2. *Game design issues*: Game designers should take into account the diversity of users, as it seems that game design affects the learning efficiency and outcome for different learning characteristics like cognitive style. New serious games can be developed incorporating game mechanics that suit different cognitive styles. For example, some games could rely on showing to the users numbers, equations and statistical data (best for Judgers) whereas other should start with a short story or a case study or an example of a problem that the user needs to target (best for Sensors and Feelers). This means that Judgers will do best when they are presented with the problem 2+2=? and Sensors/Feelers would like to see a sentence like: "Imagine you have 2 apples and you find 2 more...how many do you have in total?". Stories and real life problems incorporated in games can be very helpful especially for Feelers, since they appreciate opportunities for affective learning. Furthermore, games can use symbols and abstraction to accommodate students with high intuition but also present facts and examples for Sensor students.

3. *Curricula issues*: The curriculum design that includes games for CSE should offer both bottom-up (Sensors) and top-down (iNtuition) learning opportunities. In that light, games can be used both as specific examples and to present the larger picture. Finally, the structure of the learning activities followed in the classroom and the purpose of the games used, needs to be clear to the students, since it supports Thinkers.

Concluding, as the society comes to grips with the information revolution, the ability to deal basic programming concepts becomes an increasingly important skill. GBL methods could support children to acquire such programming skills much easier than the traditional teaching methods. The competence to appropriately apply GBL to different students' needs and particularities such as cognitive style can be a critical aim for education. However, the present work is a correlational study, meaning that causality is not implied. In that light we cannot imply that any factor such as GBL or even novelty effect (improvement simply due to the fact that people are exposed to a new technological condition) can lead to increased performance. The importance of the present work lies in the fact that other influential factors emerged such as cognitive style. We hope the present study resulted in important findings since it sheds light into the phenomenon of individual learning of programming concepts through games.

9.8.2 Limitations

The sample size of our study was relatively small. A sample size of 77 participants gives a margin of error of 7.8% (Confidence Level: 95%). The margin of error could be reduced by running this study with more participants (e.g. with 150 participants, the margin of error drops to 5.6%). With a broader sample size evidence to further support these conclusions could be given and allow these results to be further generalized beyond the scope of this study.

In our study, we used a post questionnaire to reveal students' attitudes towards games and GBL. Single item questions that were used may have limited reliability and validity. The extent to which a measure comprises reliable content sets the upper boundary for the correlation of that item with any source of validation. Items with low reliability are restricted in their ability to correlate with everything. Thus, with only a single item, we cannot be sure how reliable that item is, so we are inherently limited in that regard.

A pre-test analysis of the computing skills would have made the results more robust. However, we did not have this choice, due to school regulations and limited access to students. The procedures of accessing students in Greek schools are very restricting, not allowing for the option of running pre-tests.

The attitudinal post-test seems to confirm the popularity of the intervention, which could be also due to the novelty effect. However, it is not clear whether the attitudes of the students were changed by the activity, which would be the subject of a longitudinal study to investigate.

Finally, there may be possible biases through the specific games used. The game type might be also an affecting factor and other games might produce different results. For example, some students might be better at puzzle games whereas others might prefer adventure games. For this reason, we cannot generalize the findings for all types of games but we can only claim that with the specific gaming activities, cognitive style seems to correlate with different students' performance. In addition, the games used here do not support all possible programming phases but only specific ones like problem preparation, coding and debugging (they do not support program design). In that light, the results presented here cannot be generalized to GBL in CSE but they only apply to the specific programming activities used here.

9.8.3 Future Work

Similar GBL activities can be repeated in the future, and could especially focus on approaches for introducing game activities to different students' cognitive styles. Whilst this study reported on participants solving simple game-coding activities, it would be valuable to run this study to test the guidelines against a wider range of programming games and abilities.

Additionally, a follow-up study could focus on studying students' role, preferences and behavior, whilst targeting on a wider range of participants from multiple schools. We also believe that additional qualitative data (e.g., structured interviews with the students) is necessary to explore the relationship between the attitudes of the students and the knowledge acquisition. With more in-depth qualitative-based studies we would be able to identify the exact features of the game which help students increase their enjoyment, engagement and learning performance. Furthermore, mixed methods [691] can also be used, since the phenomenon of student performance is multifactorial and gaming is probably one affecting factor (as the significant correlations found here imply). However, other factors might be of equal importance and under that light future studies could focus on experimental analysis of the case, providing findings that allow the identification of causal factors.

Also, it might be certain cognitive styles that affect some programming phases (problem preparation, program design, coding, and debugging) but not others [651]. In future work, we could investigate students' attitudes and performance in particular programming phases.

Furthermore, it would be interesting to compare traditional teaching approaches with regards to cognitive style and GBL. In our future we will see whether GBL is better suited for specific cognitive styles whereas traditional teaching is better for others. We are also planning to test students' performance in school settings in correlation with their cognitive style and their performance on games. Possible correlations in students' school performance and games performance could be found to suit some students better.

Finally, further study could observe pair programming through games in correlation with students' cognitive style. Students could work in pairs and their performances in games could be investigated in relation to their learning characteristics. Some cognitive styles might co-operate better than others as far as GBL is concerned.

It is clear that there are several aspects of this research that could be investigated further. Cognitive style and GBL seem to be mutually correlated. We hope, this study's findings pave the way to studying programming learning through games, taking into account students' needs and personal learning characteristics like the cognitive style. Educational groups have been working for years to build the infrastructure needed to support CS both inside and outside the school environment. A carefully designed gaming curriculum, that takes into consideration the deeper personality structures of the learner, could be a basic part in this effort.

9.9 Elements in the ATMF

This section presents the elements of the ATMF that were affected by this research study (RS5). Figure 92 shows the teaching methods that were used. Non textual programming through games was implemented that led to further research of personalized learning.

ATMs		
Element 2		**Teaching**
	Section 1	Peer Learning
	Section 2	Problem-Based Learning
	Section 3	Project-Based Learning
	Section 4	Studio-Based Learning
	Section 5	Inquiry-Based Learning
	Section 6	POGIL
	Section 7	Team Learning
	Section 8	Game-Based Learning
	Section 9	Educational Robotics
	Section 10	Non-Textual Programming
	Section 11	Contextualized Learning
	Section 12	CS Unplugged
	Section 13	Subgoal Learning
	Section 14	Programming Puzzles
	Section 15	Extreme Programming
	Section 16	Program Visualization
	Section 17	Competency-Based Learning
	Section 18	Social Networks Learning
	Section 19	Emerging Technologies

Figure 92. ATMs used in the RS5.

Figure 93 displays the elements of each component that were used in this study. From the first dimension, the "Non-formal education" element from component 1a is mentioned. Students worked in school place with school's resources but outside school's hours and program.

Dimension 1: Context		
Component 1a		**Education Level**
	Element 1	Formal Education / Type of School / Level
	Element 2	Informal Education
	Element 3	Non-formal Education
Component 1b		**Environment (physical)**
	Element 1	Infrastructure (availability)
	Element 2	Supporting Resources
Component 1c		**Educational system**
	Element 1	Policies
	Element 2	Compulsory Curriculum
	Element 3	CS standards
	Element 4	General public attitudes
	Element 5	Time restrictions

Dimension 3: Content		
Component 3a		**Learning**
	Element 1	Subject
	Element 2	Objectives
	Element 3	Short-term & Long-term Planning
	Element 4	Expectations for learning
	Element 5	Relations to other subjects / disciplines
Component 3b		**Teaching**
	Element 1	Lecture-based
	Element 2	ATMs (see Figure 33)

Dimension 2: Participants' Characteristics		
Component 2a		**Learner related**
	Element 1	Age
	Element 2	Gender
	Element 3	Cultural background
	Element 4	Socioeconomic status
	Element 5	ICT use
	Element 6	ICT experience
	Element 7	Attitudes
	Element 8	Personality traits
Component 2b		**Instructor related**
	Element 1	Age
	Element 2	Gender
	Element 3	Teaching experience
	Element 4	Academic characteristics
	Element 5	Skills - Abilities
	Element 6	Attitudes
	Element 7	Personality Traits

Dimension 4: Evaluation		
Component 4a		**Learner related**
	Element 1	Knowledge / Performance
	Element 2	Motivation
	Element 3	Skills
	Element 4	Metacognition
	Element 5	Beliefs
Component 4b		**Instructor related**
	Element 1	Assessment Criteria / Method
	Element 2	Monitoring of Learning (Learning Analytics)
	Element 3	Feedback to Learners
	Element 4	Long-term Evaluation
	Element 5	Teaching Evaluation

Figure 93. ATMF Elements of RS5.

From the second dimension there were four learner related elements mentioned. ICT use (element5) and experience (element 6) were investigated in correlation with other factors such as students' attitudes (element 7) and cognitive style (element 8 about personality traits). Since personality seems to play an important role for learner it was also assumed that it might be important for instructors too (element 7, component 2b, dimension 2).

Moreover, from the third dimension, the element of "Expectations for learning" from component 3a is highlighted. The hypotheses of the RS5 support this part.

Finally, from the fourth dimension, four elements from component 4a were highlighted. About Element 1, the students' performance and gained knowledge were assessed. Their motivation and developed skills were also evaluated (elements 2 and 3). In addition, their beliefs (element 5) were investigated. From component 4b, the first element about assessment method is mentioned, since an empirical evaluation method was used.

9.10 Summary

This study investigated students' attitudes from gaming activities to reveal the quality of their learning experience and correlated it with their cognitive profile to reveal potential differences. In addition, through an empirical way it was revealed the GBL effectiveness and next it was compared to students' cognitive styles. Cognitive style was found to be a significant learning characteristic that should be taken into consideration when using digital games to learn programming.

Part IV - Closing

Chapter 10

An Investment to Knowledge, Pays the Best Interest
Benjamin Franklin

Conclusions

This final chapter summarizes the research work presented in this thesis and discusses its achievements and limitations along with suggestions for future work. This dissertation has presented a framework for effective implementation of alternative teaching methods in CSE. For the conception of the framework, every known method in CSE has been analyzed. Moreover, the thesis has provided empirical evidence in Greek settings that alternative teaching provides 1) motivation for learning and 2) better learning outcomes. Both previous work and empirical research shaped the Alternative Teaching Methods Framework (ATMF).

10.1 Summary of findings

This thesis is primarily based on three research questions as proposed in Chapter 1: (1) what are the alternative teaching methods (ATM) that can be used in CSE, (2) which are the key benefits of implementing ATM in Greek education and (3) how can ATM be used effectively in CSE. Hence, the main goal was to design an educational framework based on ATM and test it in the Greek education. For that reason, five empirical research studies were conducted.

10.1.1 Characteristics and Strengths of ATM

The first research question was answered through an extensive literature review (section 3.4) about alternative teaching methods, approaches, strategies and tools used in CSE. There are several methods used by educators and researchers in order to deal with the complexity and the needs of various cases. The use of them though could be adapted in more approaches with similar features. The following table describes every known teaching method that can be applied within CSE and highlights the key benefits.

Table 48. Alternative Teaching Methods that have been used in CSE.

Teaching Method - Approach	Learning Approach - Objectives	Knowledge Areas - Related methods	Benefits	References
Peer Learning (**PL**)	In **PL** students interact with other students to attain educational goals. It is actually learning from each other. Collaborative Learning (**CL**), is learning with each other. In Peer Tutoring (**PT**), advanced students take on a limited instructional role. In Peer Instruction (**PI**), students consider the problem on their own and contribute their answers in a way that the fraction of the class giving each answer can be determined and reported. In Peer Assessment (**PA**), students grade assignments or tests of their peers based on a teacher's benchmarks.	PT, PI, PA, CL	• increased performance • organizing and planning skills • working collaboratively with others • giving and receiving feedback • evaluating their own learning • increased motivation	[141, 152, 153, 155, 157, 158]
Project-Based Learning (**Project-BL**)	In **Project-BL** students explore problems that are important to them and find answers through the completion of a project. A project-based classroom allows students to investigate questions, propose hypotheses and explanations, discuss their ideas, challenge the ideas of others, and try out new ideas. Project-BL is an approach where the goals are set. It is also quite structured in the way that the teaching occurs. In addition, project-based follows general steps while problem-based provides specific steps.	Problem-BL, CL, Team Learning	• higher scores • deeper understanding • apply knowledge to new situations • engaging, promotes active learning • higher order thinking skills • success skills for college, career • responsibility, confidence, work in teams, communicate ideas • use of technology effectively • teaching is more enjoyable	[170], [171], [172-174], [175], [176]
Problem-Based Learning (**Problem-BL**)	**Problem-BL** is learning by solving a large, real-world problem. In PBL students undergo a complete learning journey that involves finding a problem, gathering	Project-BL, CL, Team Learning	• student-centered - fosters active learning, improved understanding and retention and development of lifelong learning skills	[160, 163], [162], [164], [165], [166]

	evidence, defining it, proposing different solutions and then arriving at a solution. Problem-based learning focuses on the problem and the process, while project-based learning focuses on the product.		• develop generic competencies - skills and attitudes • facilitates an integrated core curriculum • motivation, deep learning • engagement • constructivist approach - activate prior knowledge and build on existing conceptual knowledge frameworks	
Studio-Based Learning (**SBL**)	**SBL** is an approach applied to the teaching and learning computing that emphasizes the iteration of the skills of computational thinking, critical analysis, collaboration, and communication. Students solve complex design problems collaboratively and present their solutions to peers and instructor(s) for review followed by revision(s) reflective of the review process.	Team Based Learning, CL, Project-BL, Design oriented Learning	• improves student outcomes • stronger sense of community • improved critical thinking skills • prevention of premature convergence to incomplete knowledge • engaged, invested and motivated • presentation skills • enjoyment, motivation and interest in CSE, and become more proficient practitioners of computational problem-solving by getting experience on learning from the design exercises	[179, 188], [180], [183, 184], [185, 186], [187]
Inquiry-Based Learning (**IBL**)	**IBL** is an approach in which students are actively engaged in the learning process by asking questions, interacting with the real world, and devising multiple methods to address the questions. Learners in IBL are guided by questions that lead to gathering of evidence, formulating explanations from the evidence, communicate, and justify the explanations. The teacher plays the role of a cognitive guide and a facilitator in the process.	Problem-BL, Team Based Learning, CL	• learners take active role in their education • provides with opportunities to engage in complex tasks that otherwise would be beyond their current abilities • scaffolding learning process	[192, 193], [199], [200]
Process Oriented Guided Inquiry Learning (**POGIL**)	In **POGIL**, teams of learners (typically 3-5) work on scripted inquiry activities and investigations designed to help them	CL, Inquiry Based Learning	• improves student outcomes • active learning	[205-209], [209, 211, 212], [213],

	construct their own knowledge, often by modeling the original processes of discovery and research. The teams follow processes with specific roles, steps, and reports that encourage individual responsibility and metacognition. POGIL activities and processes are designed to achieve specific learning objectives. The instructor serves as a facilitator, not a lecturer.		• discover and understand even the difficult programming concepts • communicate and work in teams more effectively • skills such as communication, teamwork, critical thinking, and problem solving • helps instructors overcome problems • effective in incorporating voice and higher order thinking to better teach for diversity and attract students to take more CS courses in higher education.	[214], [215], [216]
Team Learning (**TL**)	**TBL** is a structured method that can help instructors improve student learning and energize their classrooms. Team Based Learning (TBL) is an approach to course design that takes advantage of research into what helps students learn. Students work in permanent teams and course meetings are organized around application exercises. Courses are divided into units, and students are held responsible for the core reading in a beginning-of-unit test taken both as individuals and as teams. Collaborative learning (**CL**) is commonly when groups of students work together to search for understanding, meaning, or solutions to create an artifact or product of their learning.	CL, IBL, PL	• cognitive constructivism, where peer discussion leads to improved conceptual understanding • teamwork and deeper understanding that are promoted among team members with varying levels of prior knowledge • creativity, by transforming knowledge into a product that is fulfilling for the student • communication skills and critical thinking, by public oral presentations	[221], [224], [225], [226], [227]
Pair Programming (**PP**)	In **PP** Students work as a pair together on one workstation. The pair is made up of a driver, who actively types at the computer or records a design; and a navigator, who watches the work of the driver and attentively identifies problems, asks clarifying questions, and makes suggestions.	TL, CL	• higher performance • increased motivation • higher quality of programs • enjoy programming • easier for instuctors	[230, 237, 692-694]
Participatory Learning	Participatory methods are those which draw the student into the learning process and thus	TL, CL, PD	• personal involvement • encourage better retention of	[240], [245-250], [251]

	student becomes a participant who articulates in some way what is learned. They are designed for small groups of participants. Typically, participatory methods are considered brainstorming, role playing, workshops etc. Participatory Design (**PD**) The PD method is most used within the HCI field. In the PD process, the designer engages in analysis activities that provide an insight into the user's conceptual model or mental model of the tasks for the system that is being targeted for development.		knowledge • active participation and interest in the learning process are essential elements for success	
Game Based Learning (**GBL**)	**GBL** refers to teaching and learning activities carried by adopting games. In GBL methods the instructor uses tips, techniques, and tools that apply the principles of game design to the learning process. Digital game-based learning (**DGBL**) is an instructional method that incorporates educational content or learning principles into video games with the goal of engaging learners. Applications of digital game-based learning draw upon the constructivist theory of education. **Gamification** is the concept of applying game mechanics and game design techniques to engage and motivate people to achieve their goals.	Problem-BL, CL	• support learning in a rather positive way • they add a pleasant experience during the learning process • give extra motivation for learning. • effective and useful • games in the curriculum help students experience how GBL can contribute to teaching and learning • supports introductory programming	[252], [256], [257], [263-266], [267, 268], [254, 255, 269], [267, 268, 270, 271], [272].
Educational Robotics (**ER**)	**ER** is about the set of educational activities that support and strengthen specific areas of knowledge and skills developed in students through the design, creation, assembly and operation of robots. The primary objective of ER is to provide a set of experiences to facilitate learners' development of knowledge, skills and attitudes for the design, analysis, application and operation of robots	Problem-BL, CL, STEM education	• improves learning • motivates and facilitates the instruction of computer programming • promotes creativity, participation and supports teamwork. • direct output of programs and not just write code or see virtual characters on a screen. • engage with CS concepts and	[98], [294], [295, 296], [297], [298], [299], [300], [301]

			programming in a fun and creative and they increase interest levels, motivation, problem-solving and student collaboration • support constructivist learning and provide the opportunities for the personal construction of knowledge and meaning • learners do not learn from technology, but they learn with technology	
Non-textual Programming	**Non-textual programming** is about tools and programming languages to help novices learn to program like: narrative tools, visual programming tools, flow-model tools, multimedia and tiered language tools. Particularly, the programming languages that have been developed to mitigate the various difficulties of programming education are known as Educational Programming Languages (**EPL**) In Visual Programming Languages (**VPL**) users create programs by manipulating program elements graphically rather than by specifying them textually. **Tangible computing** deals with tangible objects that are interfaced with computers.	Problem-BL, PP, VPL, EPL, tangible programming, introductory programming	• learn CS concepts, easily • develop higher order thinking skills such as algorithmic problem solving • learning by example, and analytical reasoning strategies • programming can make it easier to see the big picture • help maintainability, since students can more easily see what the program it does • simplifying the programming language syntax • matching more closely to the way that children describe the behavior of programs	[98, 335], [316, 317], [346], [347], [348], [336],
Contextualized Learning	A **contextualized computing** course is one in which one or more application domains provide the motivation for learning CS content and inspire the design of learning activities; these domains may be, and often are, external to CS itself. Contextualized Learning always take place in a context, whether we build on that context or ignore it. We can either recognize this, use context as a tool, and make it highly effective in supporting student learning or ignore it and hope for the best.	Problem-BL, introductory programming	• following in the constructivist tradition, contextualized learning may increase learning and motivation • may increase the number of students who are attracted to CS and seek to remain in the field • it gives a good background for building programming skills • captures learners' imagination and make them more creative	[355, 356], [357, 358], [359, 360], [361]

	Media Computation (**MC**) is an approach to that explains how digital media are manipulated. For example, students learn about loops by changing all the pixels in a picture to compute a negative image, or all the samples in a sound in order to decrease the volume. Students may learn about conditionals by removing red eye in the image without changing any other colors, or changing only part of a sound.		• can help improve gender balance in computing	
CS Unplugged	**CS Unplugged** is a set of active learning activities designed to introduce fundamental computer science principles without the use of computers. This approach introduces students to CT through concepts such as binary numbers, algorithms and data compression, separated from the distractions and technical details of having to use computers. The unplugged approach presents fundamental concepts in several CS areas such as networks, artificial intelligence, graphics, information theory, Human Computer Interfaces, and of course programming languages.	Problem-BL, CL, Team Learning	• no need for computers • engages to CS without programming • the activities allow students to discover answers for themselves • students realize that they are capable of finding solutions to problems on their own, rather than being given a solution to apply to the problem • promotes CS to students as an interesting, engaging, and intellectually stimulating discipline • the activities capture learners' imagination and address common misconceptions about what it means to be a computer scientist	[367], [368], [369]
Subgoal Learning	**Subgoal learning** has been used to help learners recognize the fundamental structure of the procedure being exemplified in worked examples. Worked examples give learners concrete examples of the procedure being used to solve a problem. **Subgoal labeling** is a technique used to promote subgoal learning and is used in the fields of cognitive science and educational psychology. The technique is about giving a name to a group of steps, in a step-by-step description of a process, to explain how the group of steps achieve a related subgoal.	guided instruction, introductory programming	• promotes deeper processing of worked examples • improves retention and transfer • improves programming learning • produces higher learning gains and better problem solving performance • promotes self-explanation • helps learners identify the structural information from incidental information and can reduce cognitive load because the	[370-372], [373], [374], [375, 376], [377], [378], [379]

			learner has fewer possible problem-solving steps to focus	
Parson's Programming Puzzles	**Parson's Programming Puzzles** is a method that provides practice with basic programming principles in an entertaining puzzle-like format. Careful design of the items in the puzzles allows the instructor to highlight particular topics and common programming errors. Since each puzzle solution is a complete sample of well-written code, use of the tool exposes students to good programming practice. Parson's puzzles are simplified code construction assignments where the lines of code are given in the wrong order and the task is to sort and possibly select the correct lines.	GBL, introductory programming	• engaging learning environments • immediate feedback • automatic assessment • effective and efficient learning approach than writing especially for time-strapped secondary teachers • considerably easier and less subjective to mark • better for weak students • helps with syntactic problems of programming languages to recognize patterns more easily	[381], [382], [383], [384, 385]
Extreme Programming	Extreme Programming (**XP**) is a computer software development approach that credits much of its success to the use of PP by all of their programmers, regardless of experience. XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software. It is one of the Agile software development methodologies. The method of **eXtreme Teaching** is inspired by XP and highlights the importance of values such as feedback, communication, respect and courage.	eXtreme Teaching, PP, agile methodologies, software development, CL	• has all the PP method benefits • improves the way learners think about programming and program design • increases dialogue between the involved instructor and learners • learners feel more committed to the course as they become more involved in it • encourages active learning • enhances motivation • gives a solid basic of iterative development, configuration management, and team communication	[236], [388], [389], [390], [392]
Program Visualization	Program Visualization or else Visual Program Simulation (**VPS**) is a pedagogical technique for introductory programming education. In VPS, a student takes on the role of the computer as executor of a given program. The student uses a visualization of an abstract computer, a notional machine, as an aid to illustrate what the computer does as it processes the program. The goal of the VPS activity is to help the student learn about	introductory programming	• provides an interactive worked-out example of program writing • judicious use of examples helps manage learners' cognitive load during the learning process • understand how computers execute programs, and that in turn relates to how one reads and writes	[396], [398], [399]

	programming in general and about specific programming concepts.		computer programs as a programmer • help learners discern the crucial dynamic aspect to programs and programming, and to construct a better mental model of the mechanisms underlying program code • helps novice programmers across a key early threshold	
Competency Based Learning (**CBL**)	CBL is an approach to teaching and learning that gives importance in concrete skills than abstract learning. These strategies include among others situated learning, online and blended learning, flipped learning and learning by taking account personal characteristics. CBL is examined by a more abstract point of view, where teaching and learning can use different techniques and tools for the learners' service. Therefore, the most important characteristic of CBL is that it measures learning rather than time. Moreover, learners have different objectives and predispositions. Hence, an optimal learning path for one learner is not necessarily the same for the others. That is the key element of Personalized Learning (**PL**) method which seeks to accelerate student learning by tailoring the instructional environment to address the individual needs, skills and interests of each student. Students can take ownership of their own learning, while also developing deep, personal connections with each other, their teachers and other adults. Blended Learning (**BL**) is a formal education program in which a student learns at least in part through delivery of content and instruction via digital and online media with some element of student control over time, place, path, or pace.	PL, FL, BL, MOOCs	• provide flexibility in the way that knowledge is earned, and provide students with personalized learning opportunities • particularly suited to CS due to the nature of the discipline where well-defined, measurable outcomes can be designed and implemented • lead to engagement and motivation because the content is relevant to each student and tailored to their unique needs. • greater understanding and better student outcomes because the pace of learning is customized to each student • increased student retention and completion rates, particularly when prior learning can be applied to the learning progress • learning resources, and assessments are aligned to well-defined goals and therefore, instructors improve their ability to understand learners' competencies and learning achievements	[400], [405], [406, 407], [408], [409], [419], [420]

	Flipped Learning (**FL**) is an instructional strategy and a type of blended learning that reverses the traditional educational arrangement by delivering instructional content, often online, outside of the classroom. It moves activities, including those that may have traditionally been considered homework, into the classroom. In a flipped classroom, students watch online lectures, collaborate in online discussions, or carry out research at home and engage in concepts in the classroom with the guidance of the instructor. Online environments **MOOCs** (=personalized online courses/distance education).			
Social Media in Education	Tools as Facebook in Education	Web 2.0,	• lead to learning, since they support easy exchange of information, communication of learners, social connections • build better relationships between the university, students and staff • student behavior is culturally affected • desirable attributes that could be beneficial to learning • support socialization and communication of users - potential learners • can change the very nature of learning, since technology is not a neutral medium	[102, 436-439], [444, 446], [448-451], [452], [453, 454]
Emerging Technologies	**Mobile technologies** offer the opportunity for learners and instructors to communicate easier and to call on information and knowledge. **Tablet computing** reveals a strong uptake in education since it has become a pervasive	mobile technologies, tablet computing, IoT	• portability, the touch screen feature, ease of use, long battery life, and affordable hardware and software • growing trend towards personalization of learning and	[459], [461], [462], [463], [464], [466]

	part of everyday life in much of the world. Internet of Things (**IoT**) is about a world in which all electronic devices are networked and every object, whether it is physical or electronic, is electronically tagged with information pertinent to that object.		increased flexibility and access • take ownership of their learning • more convenient way of learning for everyone	

10.1.2 Research studies of this thesis

The ATM framework provides a cohesive model, well founded on constructivist and collaborative learning theories, which is an applicable teaching pedagogy, especially suitable for every student cohort.

The empirical findings of Chapter 5 (RS1) confirmed the positive effects of PL in promoting learning and motivation. In particular RS1 compared learning results of alternative to traditional ways of teaching CS in Greek secondary education. The learning from Peer Teaching and Collaboration techniques was assessed and students' attitudes towards the alternative teaching were researched. In addition, teachers' opinions about that way of teaching and learning were explored. The study to was used as an input during the design of the ATMF and provided evidence that the specific ATM had better learning results and increased motivation by learners.

The empirical findings of Chapter 6 (RS2) confirmed the positive effects of SNs in promoting learning and motivation. In particular this observational study revealed tendencies of social networks' use in higher education in a specific educational and cultural environment. Facebook was used as a teaching tool in Greek higher education and ways that SNs can be used in teaching and learning were investigated. The study reported that students were highly motivated to participate in the lesson through the Facebook page although there seem to exist some personalities, cultural and gender differences about the usage.

The empirical findings of Chapter 7 (RS3) confirmed the positive effects of GBL in promoting basic programming principles to young children. Empirical findings derived from this analysis provided valuable information games and pupils' satisfaction and willingness to use them for acquiring programming knowledge. In addition, Pair Programming method was students' choice for learning programming. However, the short activity of the RS3 did not had long term results and motivation at children and thus the element of long-term planning was added to the ATMF.

The empirical findings of Chapter 8 (RS4) demonstrated the importance of referring to alternative learning through the instructor's views. In particular Educational Robotics through a national competition was researched. The study investigated the benefits of students' involvement with robotics about skills, motivation and learning through the teachers' eyes. Additionally, it researched weather ER should be introduced in the Greek compulsory curricula. The results showed that there are numerous benefits for students: they increased their collaboration, problem solving and creativity skills; understand STEM concepts about CS and engineering and especially gaining programming knowledge. In addition, most of the teachers considered that ER should be part of the compulsory curriculum. Therefore, the ATMF was enhance with the instructors' perspective about the learners knowledge and motivations.

The empirical findings of Chapter 9 (RS5) identified different personality traits and especially cognitive style as an important factor in programming learning through serious games. This study investigated students' attitudes from gaming activities to reveal the quality of their learning experience and correlated it with their cognitive profile to reveal potential differences. In addition, through an empirical way it was revealed the GBL effectiveness and next it was compared to students' cognitive styles. Cognitive style was found to be a significant learning characteristic that should be taken into consideration when using digital games to learn programming. The study to was used to finalize the ATMF and provided evidence that personality traits may affect both teaching and learning.

Finally, the limitations of each study in the thesis were discussed respectively in each chapter.

10.1.3 The ATMF

The ATMF provides a cohesive model, well founded on constructivist and collaborative learning theories, which is an applicable teaching pedagogy, especially suitable for every student cohort. It highlights what is required to build a comprehensive and effective teaching approach with alternative methods in the center (Figure 94).

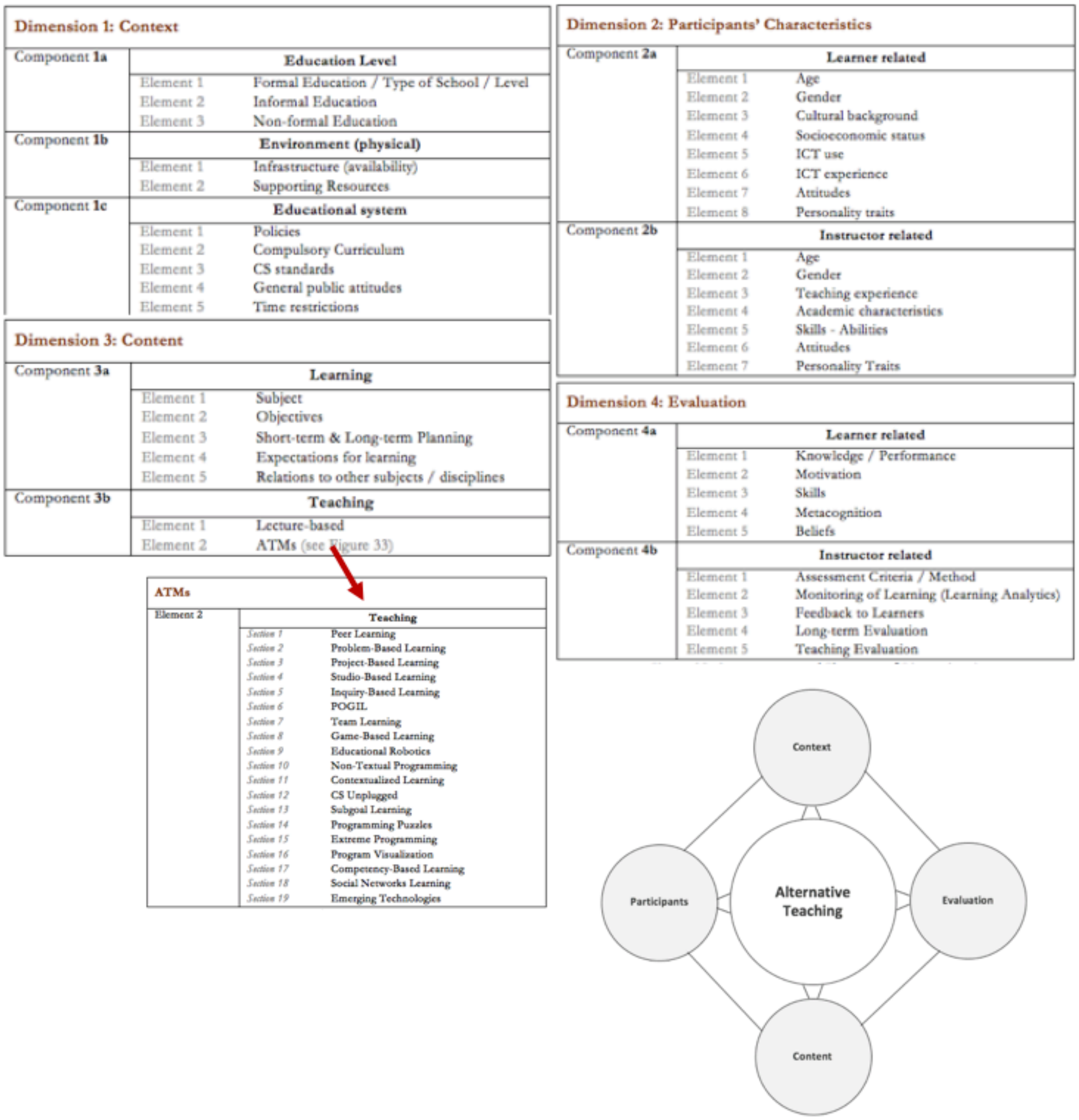


Figure 94. The framework of this thesis (ATMF).

The ATMF outlines the characteristics of alternative teaching, and the culture that needs to be in place for successful implementation to every context and content. The proposed framework is about student-centered teaching pedagogy trying to address issues around learning and teaching of CS concepts. Following a student centered approach this framework transforms the authoritative role of the instructor to a guide and empowers students with active learning and leadership skills. The ATMF is implemented based on the perspective that learning is a socially embedded cognitive process and knowledge is socially constructed through interaction and activity with others. The main objective of the ATMF is to promote motivation and enhance learning.

This thesis proposes that this model can be used in examining instructors' CS lessons of any level and any context and in designing experiences for teachers on the integration of student-centered practices in CS teaching. Use of this framework motivates and engages students with an enriching and enjoyable learning experience, assists in developing a deeper understanding with improved cognitive skills, empowers students with ability to work in team and collaboratively solve problems, and transforms the whole learning process into an interactive, knowledge sharing experience. The various benefits of the ATMF as evidenced from the empirical studies conducted within this thesis are discussed through chapters 5-9.

The proposed framework for teaching with the use of alternative methods can make teaching both more effective and more efficient, by helping instructors create the conditions that support student learning and minimize the need for revising materials, content, and policies. While implementing these principles requires a commitment in time and effort, it often saves time and energy later on.

10.2 Contributions and Implications

This thesis provides a number of contributions:

- A high level description/review of alternative teaching methods that are used in CSE, identifying strengths and areas for improvement.
- Empirical evidence that alternative teaching methods in Greek education provide:
    - Extra motivation for learning.
    - Better learning outcomes.
- A conceptual framework based both on previous theoretical frameworks and models as well as empirical guidelines, for the effective use of alternative teaching methods in CSE with the following characteristics:
    - The framework does not teach computer science; rather, it focuses on computer science teaching.
    - The framework is not limited to the teaching of a specific computer science curriculum, neither is it limited to the teaching of a specific programming language nor to a specific programming paradigm.
    - The framework can be adapted to the teaching of any CS topic in any context and any level, from elementary school through high school to the university level.

To summarize, this thesis describes a framework to teach CS concepts through alternative teaching methods in a way that appeals to a broad audience, engages its users, and shows measurable learning outcomes.

10.3 Future Directions - Final Remarks

In addition to future studies addressing the empirical research studies of this thesis, findings inform researchers in the CSE fields to seek alternative teaching and learning approaches to further enhance cognitive and social development for young learners. I identified the following aspects as research extensions from this thesis:

Conduct more experiments in Greek education with different ATMs and with more students. Moreover, I intend to research correlations of personality with different teaching and learning methods and also examine other personality traits like intelligence type or emotional type in correlation with learning various CS concepts. In addition, I aim to provide a whole analysis of the Darmstadt model [39] (described briefly at the RS5), in order to be able to compare the empirical results that the ATMF was based, with different countries within the CSE field. Finally, there is scope for extending the use of ATMF in other science and technology courses or where the student cohorts are diverse in their context, background, or aptitude.

The challenge to being an effective CS educator is to be authentic and complex enough to maintain motivation, and to use methods to support student success and make learning more efficient. Anderson et al. [362] understand that motivation is a critical part of learning. This thesis presents an effective way to help students sustain their motivation and support their learning. However, good instruction may not be just a matter of choosing the most efficient methods of teaching. The explosion of learner diversity means instructors need knowledge and skills to customize learning for learners with a range of individual differences. Instructors need to recognize that all learners bring to their learning varying beliefs, personalities, experiences, abilities, talents, and prior learning, as well as language, culture, and family and community values that are assets that can be used to promote their learning. To do this effectively, instructors must have a deeper understanding of learners' frames of reference, the potential biases in these frames, and their impact on expectations for and relationships with them so as the ATMF states, to provide multiple approaches to learning for each context, content and student.

As a result, this thesis provided theoretical and empirical exploration of alternative teaching in CSE concerning to altering the perception of reality, and thus illuminated directions of future studies. I hope that it inspires the message that CS learning and teaching processes can be enjoyable, collaborative, challenging, and motivating, and by providing this idea, learners' interest in CS learning on all levels will be increased.

10.4 Thesis Publications

Peer - reviewed international Journal publications:

1. Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2016). How do different cognitive styles affect learning programming? Insights from a Game Based Approach in Greek Schools. *Transactions in Computing Education*. ACM. 17, 1, Article 3, 25 pages. DOI: http://dx.doi.org/10.1145/2940330.

2. Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2016). Students teach students: Alternative teaching in Greek secondary education. *Education and Information Technologies,* 21(2), 373-399. DOI=http://dx.doi.org/10.1007/s10639-014-9327-7

3. Antoniou A., Theodoropoulos A., Christopoulou K., Lepouras G. (2014). Facebook as teaching tool in Higher Education: A case Study. *International Journal of Advances in Social Science and Humanities,* 2(3), 43-56.

Peer - reviewed papers in international Conference proceedings:

1. Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2016). Educational Robotics in the Service of CSE: A Study Based on the PanHellenic Competition. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (WiPSCE '16), Münster, Germany. ACM, New York, NY, USA, 84-87. *DOI*=http://dx.doi.org/10.1145/2978249.2978262
2. Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2015). The Little ones, the Big ones and the Code: Utilization of digital educational games in primary school pupils. In *Proceedings of the 7th Conference on Informatics in Education (7th CIE2015),* Piraeus, Greece, 49-59.

Peer - reviewed papers in national Conference proceedings:

1. Theodoropoulos, A., Lepouras, G. (2016). Alternative teaching methods in Informatics. In *Proceedings of the 1st PanHellenic Conference, with international participation: Digital Educational Content and E-learning 2.0 (ELOER2016)*, Korinthos, Greece, 33 (in Greek).

Appendices

A. Experiment Materials of RS 1

A.1 Students Test and Questionnaire (in Greek)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

Τμήμα:

Όνομα:

Ημερομηνία:

Θέλουμε τη βοήθειά σου!

Το Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πελοποννήσου, κάνει μία έρευνα για τον τρόπο διδασκαλίας της πληροφορικής στα σχολεία. Στις επόμενες σελίδες θα βρεις ερωτήσεις για το μάθημα των υπολογιστών στην καθημερινή μας ζωή και ερωτήσεις για τον τρόπο που έγινε το μάθημα. Απάντησε όσο καλύτερα μπορείς! Οι ερωτήσεις θα αξιολογηθούν ανώνυμα από τον καθηγητή σου! Αν έχεις οποιαδήποτε ερώτηση, οι ερευνητές θα χαρούν να σου απαντήσουν.

Σε ευχαριστούμε πολύ!

Students Test

1. Δώσε δύο παραδείγματα χρήσης του υπολογιστή στην καθημερινή ζωή.
    -
    -

2. Γιατί λέμε ότι ο υπολογιστής είναι ένα νέο μέσο επικοινωνίας;

3. Μπορείτε να αναφέρετε 3 δραστηριότητες που μπορούν να γίνουν από απόσταση;
    -
    -
    -

4. Ποιες αρνητικές επιπτώσεις μπορεί να έχει η χρήση των υπολογιστών στη ζωή μας;

Students Questionnaire

Πόσο σου άρεσε το μάθημα για τις χρήσεις του υπολογιστή στην καθημερινή ζωή; (κύκλωσε ένα)

1. καθόλου

2. μέτρια

3. πολύ

Πόσο διαφορετικό σου φάνηκε αυτό το μάθημα σε σχέση με τα προηγούμενα μαθήματα της πληροφορικής;

1. καθόλου διαφορετικό

2. λίγο διαφορετικό

3. πολύ διαφορετικό

Θα ήθελες να γίνουν με αντίστοιχο τρόπο τα υπόλοιπα μαθήματα πληροφορικής;

1. όχι

2. ναι

Με τον τρόπο που έγινε η διδασκαλία, πόσο νιώθεις ότι έμαθες σε σχέση με προηγούμενα μαθήματα πληροφορικής.

1. έμαθα λιγότερα πράγματα

2. περίπου το ίδιο

3. έμαθα περισσότερα πράγματα

Σου φάνηκε περίεργος ο τρόπος που έγινε το μάθημα για τις χρήσεις του υπολογιστή στην καθημερινή ζωή;

1. καθόλου περίεργος

2. λίγο περίεργος

3. πολύ περίεργος

Σε ευχαριστούμε πολύ!

B. Experiment Materials of RS 2

B.1 Instructor's Diary of use

Course Facebook account

16-2-2012
I registered as a person (did not create a special group). Students had to send Friend Requests (not just Like the group). That gave me full visibility and access into their personal profiles. Some students mentioned that they are against facebook and some others that they will not send me a friend request, because they do not want me to access their personal lives. **To ask at the end about privacy issues.**
21-2-2012
Almost a week after the creation of the profile, it looks like students still allow me full visibility of their personal interactions. Students start posting Likes on my input. It also looks that common friends of the course's space, establish personal friendships at least on Facebook. This is something that needs to be confirmed at the end of the course. It should be also noted that this is a first year class, meaning that many students still want to establish friendships and find groups to belong to. This space might assist them in that effort. **It still remains to be confirmed.**

Date	Instructor actions	Student actions	Comments
18-2-2012	Asked a question "what do you want to do on this space" with 6 possible answers (1. To exchange information with classmates, 2. To access procedural information about the course, 3. To socialize, 4. To access course work, 5. To evaluate the course, 6. Other)	14 students chose the first answer (7 females, 8 males), 6 chose the second (3 females, 3 males) and 2 chose the third (1 male, 1 female).	Only the first three options were visible, for the next you had to click on option "3 more…". It does not look like students accessed the three last options.
18-2-2012	Added video for diverse pathways in computer science		It does not look like the video was viewed by the students.
19-2-2012	Added education to my profile (Agglika Cstuop at Department of Computer Science and Technology)	One like from a female student.	
21-2-2012	Posted reminder for bringing course books to the lecture.	Three female and three male students Liked the post. One female student also commented on it.	
23-2-2012	Uploaded a link for a software that creates stories using social media.	Two female students liked the link.	
23-2-2012	Posted information about the group projects of the course.	One Like from a female student.	
24-2-2012	Posted a video on Basic Ergonomics using a computer.	One female and two male students liked it.	
27-2-2012	Asked students if I could borrow one of their course books.	Two female students offered their books.	

3-3-2012	Wished Happy birthday on a student's wall.	Student thanked.	The social face of such networks (notifying about birthdays, etc.) helps to bridge the possible communication gap between students and instructors. It also brings the Uni life closer to their other life. **This is something that needs to be confirmed at the end of the course.**
4-3-2012	Posted an advertisement for a lecture on free will (instructor's independent lecture series). The nature of the ad was supposed to start a dialogue between the students and the instructor. The questions posted said: "would you but a product advertised by a famous singer (the name of a popular Greek singer was used)? Would you vote for your mum's favorite party? Would you kill a stranger for no particular reason? If you answered NO to any of the above, then you should attend Wednesday's lecture."	An active dialogue started between the students and the instructor. Students posted humoristic comments and songs of the singer.	The questions used in the ad, helped in: 1. Advertising the event of the lecture, 2. Braking the ice between students and instructor
6-3-2012	Posted video with art installations	3 likes from girls, 6 likes from boys. 1 comment from a girl, 4 comments from boys.	They seemed to like it. Students seem to respond positively to art. **To be confirmed.**
9-3-2012	Posted an announcement about student marks on an exercise	Two likes (one female, one male)	It is unexpected students posting likes on issues like marks…
12-3-2012	Reminder for bringing their course books to the lecture.	4 likes (3 females, one male) and many comments asking for further information (e.g. which book), notifying for problems with course book (e.g. one students does not have her course book) and other students asking if the book will be definitely needed (e.g. not to carry heavy books unnecessarily).	
14-3-2012	Asked students for a mid-term evaluation of the course.	A male student suggested that we could try to translate lyrics from English songs.	This is an interesting idea, given the fact that students are really interested in music. I should try to incorporate it in the lectures.
22-3 & 3-4	Posted funny pictures	Likes and comments from many students	using an informal medium such as FB for teaching, it is easier to use humoristic material. It seems that students respond well and their motivation might increase. It is also a more

			relaxed approach to a lecturer…
4-4	Gave course marks as Oscars	11 likes and 14 comments	This was an unexpected form of giving out marks and praising best course works. Students were positively surprised. In line with educational theory, praising good work and using reinforcements is important, even for older students, like university students. Using the Oscars made the whole approach less formal, in order not to remind them of a school environment. It seems important to use creative ways to reinforce good performance, not by simply using marks…The Oscars method also allowed for a public recognition of best works…
2-5	Students sends course work	Likes from 4 students and comments from 2	The student's work was in progress and he needed confirmation that he is working in the right way…FB gave him the opportunity to send an uncompleted work for confirmation, since it is less formal and the students can also see when the instructor in online…
19-5	Thanking for course evaluation questionnaires	10 likes and 4 comments	
19-5	Students asks a question regarding course work		
25-5	Information about final course marks	11 likes and 3 comments	
20-6	Posting a funny video about football, due the upcoming football match between Greece and Germany	4 likes and 7 comments, all from male students	The popularity of football makes it a good motivating material…
23-6	A students sends a historical football story (Ukraine Vs. Germany, 1942)		The communication with certain students went far deeper than superficial jokes and procedural information. By posting other material (not simply course material), like art, political, historic information, some students felt that we could engage in deeper communication. FB gave the opportunity for this, since during lectures there is not time for other conversations… this is a very important element. I also received comments

			on political and social views.

29-2-2012
Students after the lecture asked if they could post favorite song videos on face book.

7-3-2012
A student decided to upload his course exercise on facebook (to send it to the instructor), although there are other alternatives (i.e. eclass, instructor's email, etc.). Does this imply easier access to the instructor? **To confirm if Facebook made them perceive the instructor closer to them…**

8-3-2012
A student posted on my wall a video from his favorite football team. Another student commented really surprised that he posted this information on the instructor's wall.

General notes

The instructor's posts are divided in the following categories:

- Procedural information about the course (i.e. marks now available on eclass)
- Reminders (i.e. please, bring your course books)
- Social postings (i.e. birthday wishes)
- General interest information (i.e. art installations, new paths in computer science)
- Evaluations and feedback aspects (i.e. questions like what do you expect from this course)
- Other notifications (i.e. information about uni events, like free lectures, etc.)

Students' activities in the following categories:

- Asking questions
- Asking for clarifications
- Replying to instructor's requests
- Replying to instructor's comments
- Sending course work to instructor and other students
- Sending personal thoughts on social, political issues to the instructor

C. Experiment Materials of RS 3

C.1 Instructions to parents (In Greek)

Οδηγίες προς Γονείς

Στις 11-17 Οκτωβρίου 2014 πραγματοποιείται για δεύτερη χρονιά η **Ευρωπαϊκή Εβδομάδα Προγραμματισμού**.

Εκατομμύρια παιδιά, γονείς, εκπαιδευτικοί, επιχειρηματίες και πολιτικοί ιθύνοντες θα συμμετάσχουν σε μαζικές εκδηλώσεις και θα παρακολουθήσουν μαθήματα για να μάθουν προγραμματισμό και να αποκτήσουν συναφείς δεξιότητες.

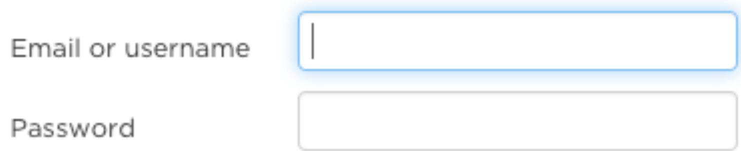
Η ιδέα είναι να προβληθεί ο προγραμματισμός, να απομυθοποιηθούν οι σχετικές δεξιότητες, και να έρθουν σε επαφή άτομα που έχουν μεγάλη επιθυμία να αποκτήσουν γνώσεις.

Στην ιστοσελίδα http://codeweek.eu μπορείτε να μάθετε περισσότερα.

Οδηγίες προς Μαθητές

Αν σας άρεσαν οι δραστηριότητες που πραγματοποιήσατε στο Πανεπιστήμιο μπορείτε να συνεχίσετε και από το σπίτι σας ακολουθώντας τα παρακάτω βήματα:

1. Επισκεφτείτε την ιστοσελίδα http://studio.code.org
2. Πάνω και δεξιά πατάμε το “**Sign in**”
3. Πληκτρολογούμε τα παρακάτω στοιχεία για να συνδεθούμε,



4. Αφού λοιπόν συνδεθείτε, **πάνω και δεξιά** (εκεί που φαίνεται το όνομά σας), πατάμε το βελάκι και επιλέγουμε “**Η πρόοδός μου**”, για να συνεχίσουμε από εκεί που είχαμε μείνει.
5. Μην ξεχνάτε στο κάτω μέρος της ιστοσελίδας μπορείτε να αλλάξετε την γλώσσα εμφάνισης σε Ελληνικά.

Ευχαριστούμε πολύ για την συμμετοχή σου!

Πανεπιστήμιο Πελοποννήσου
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

D. Experiment Materials of RS 5

E.1 Student Questionnaire (In Greek)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Σχολείο:

Τάξη - Τμήμα:

Ονοματεπώνυμο:

Ημερομηνία:

Κωδικός από Cognitive test:

Θέλουμε τη βοήθειά σου!

Το Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Πελοποννήσου, κάνει μία έρευνα για τον τρόπο διδασκαλίας της πληροφορικής στα σχολεία.
Στις επόμενες σελίδες θα βρεις ερωτήσεις σχετικές με τη δράση "Ώρα του Κώδικα" που έλαβες συμμετοχή και κάποιες γενικότερες για το μάθημα της Πληροφορικής.
Απάντησε όσο καλύτερα μπορείς!

Σε ευχαριστούμε πολύ!

1. Όνομα χρήστη στο σύστημα "Hour of Code": ....
2. Ηλικία: ....
3. Φύλο: ....
4. Προηγούμενος σχολικός Βαθμός στο μάθημα της Πληροφορικής; ....
5. Προηγούμενος σχολικός Γενικός Βαθμός (Μ.Ο. από όλα τα μαθήματα); ....
6. Πόσο συχνά χρησιμοποιείς υπολογιστή για τα μαθήματά σου; (επέλεξε μόνο 1)

Πάρα πολύ | Πολύ | Μέτρια | Λίγο | Καθόλου

7. Με ποια από τα παρακάτω προγράμματα, γλώσσες προγραμματισμού, έχεις ασχοληθεί στο σχολείο ή και αλλού (κύκλωσε όσα θες);
   1. Logo
   2. Scratch
   3. Kodu
   4. Pascal
   5. Ψευδογλώσσα (π.χ. ο Διερμηνευτής της Γλώσσας)
   6. Κάποια άλλη γλώσσα προγραμματισμού ……………………. (←γράψε ποιά)

8. Είναι σημαντικό να μάθεις Προγραμματισμό στις μέρες μας; (επέλεξε μόνο 1)

Συμφωνώ | Συμφωνώ εν μέρει | Δεν Έχω Άποψη | Διαφωνώ εν μέρει | Διαφωνώ

9. Γνωρίζεις τι είναι η Υπολογιστική Σκέψη; (επέλεξε μόνο 1)

Ναι | Όχι | Ίσως

10. Θέλεις να μάθεις περισσότερα για την επιστήμη των υπολογιστών; (επέλεξε μόνο 1)

Πάρα πολύ | Πολύ | Δεν Έχω Άποψη | Λίγο | Καθόλου

11. Πόσο σου άρεσε η εμπειρία της "Ώρας του Κώδικα"; (επέλεξε μόνο 1)

Πάρα πολύ | Πολύ | Δεν Έχω Άποψη | Λίγο | Καθόλου

12. Πόσο εύκολες βρήκες τις δραστηριότητες της "Ώρας του Κώδικα"; (επέλεξε μόνο 1)

Πάρα πολύ Πολύ Δεν Έχω

Άποψη Λίγο Καθόλου

13. Πόσες πίστες συμπλήρωσες στην; (γράψε επίπεδο και πίστα π.χ. 15-3): ……………………………

14. Πιστεύεις ότι έμαθες κάποιες βασικές αρχές του προγραμματισμού; (επέλεξε μόνο 1)

Πάρα πολύ Πολύ Δεν Έχω

Άποψη Λίγο Καθόλου

15. Θα ήθελες να επαναληφθούν παρόμοιες δραστηριότητες στο μέλλον; (επέλεξε μόνο 1)

Πάρα πολύ Πολύ Δεν Έχω

Άποψη Λίγο Καθόλου

16. Ήσουν μόνος/η στον υπολογιστή;
    1. Ναι
    2. Όχι

17. Βοήθησες κάποιον συμμαθητή σου σε κάποια πίστα;
    1. Ναι
    2. Όχι

18. Βοηθήθηκες από κάποιον συμμαθητή σου σε κάποια πίστα;
    1. Ναι
    2. Όχι

19. Προτιμάς να εργάζεσαι στον υπολογιστή μόνος σου ή μαζί με άλλον συμμαθητή σου;

Συμφωνώ Συμφωνώ εν μέρει Δεν Έχω Άποψη Διαφωνώ εν

μέρει Διαφωνώ

20. Πιστεύεις ότι μαθαίνεις καλύτερα μόνος σου;

Συμφωνώ Συμφωνώ εν μέρει Δεν Έχω Άποψη Διαφωνώ εν

μέρει Διαφωνώ

21. Παίζεις παιχνίδια στον υπολογιστή;

Πάρα πολύ Πολύ Δεν Έχω
Άποψη Λίγο Καθόλου

22. Σου αρέσουν τα παιχνίδια στον υπολογιστή;

Πάρα πολύ Πολύ Δεν Έχω
Άποψη Λίγο Καθόλου

23. Πιστεύεις ότι μπορείς να αποκτήσεις γνώσεις σε βασικές έννοιες προγραμματισμού παίζοντας παιχνίδια στον υπολογιστή;

Πάρα πολύ Πολύ Δεν Έχω
Άποψη Λίγο Καθόλου

24. Έχεις μεγαλύτερο Κίνητρο να μάθεις παίζοντας στον υπολογιστή;

Πάρα πολύ Πολύ Δεν Έχω
Άποψη Λίγο Καθόλου

E.2 MBTI Questionnaire (In Greek)

Ακολουθούν κάποιες περιγραφές προσωπικών προτιμήσεων σε δύο στήλες. Και οι δύο στήλες είναι ισότιμες. Και οι δύο είναι το ίδιο καλές. Για κάθε περιγραφή διαλέξτε ή την πρώτη ή τη δεύτερη πρόταση σε κάθε σειρά, προσπαθώντας να απαντήσετε όπως πραγματικά είσαστε στην καθημερινή σας ζωή και όχι όπως θα θέλατε να είσαστε. Σημειώστε √, δίπλα στην επιλογή σας.

E	***I***
Συνήθως έχω πολύ ενέργεια	*Συχνά νιώθω να μου λείπει ενέργεια*
Μιλάω πιο πολύ, παρά ακούω	*Ακούω πιο πολύ, παρά μιλάω*
Σκέφτομαι μεγαλόφωνα	*Κρατώ τις σκέψεις μου μέσα μου*
Πράττω χωρίς να το πολυσκέφτομαι	*Σκέφτομαι πολύ πριν από τις πράξεις μου*
Μου αρέσει να είμαι με πολύ κόσμο	*Νιώθω καλλίτερα μόνος*
Προτιμώ να είμαι στο κέντρο του ενδιαφέροντος	*Δε θέλω να είμαι στο κέντρο του ενδιαφέροντος*
Η προσοχή μου αποσπάται εύκολα	*Συγκεντρώνομαι εύκολα*
Κάνω πολλά πράγματα μαζί	*Κάνω ένα πράγμα τη φορά*
Είμαι εξωστρεφής και ενθουσιώδης	*Είμαι εσωστρεφής και προσεχτικός*
S	***N***
Δίνω προσοχή στις λεπτομέρειες	*Βλέπω τη γενική εικόνα*
Παράγω πρακτικές λύσεις	*Παράγω δημιουργικές ιδέες*
Θυμάμαι γεγονότα	*Ασχολούμαι κυρίως με καινούρια πράγματα*
Βλέπω τα πράγματα όπως είναι	*Βλέπω τα πράγματα όπως θα μπορούσαν να είναι*
Ζω το παρόν	*Προγραμματίζω το μέλλον*
Θέλω χειροπιαστές αποδείξεις	*Εμπιστεύομαι το ένστικτό μου*
Δουλεύω με σταθερό ρυθμό	*Οι ρυθμοί μου δεν είναι σταθεροί*
T	***F***
Αποφασίζω με τη λογική	*Αποφασίζω με το συναίσθημα*
Με θεωρούν απόμακρο και προσεκτικό	*Με θεωρούν ζεστό και φιλικό*
Είμαι ευθύς	*Χειρίζομαι καταστάσεις με λεπτότητα*
Προτιμώ ειλικρίνεια και δικαιοσύνη	*Προτιμώ αρμονία και συμπόνια*
Δε παίρνω τα πράγματα προσωπικά	*Παίρνω πολλά πράγματα προσωπικά*
Κίνητρό μου είναι η επιτυχία	*Κίνητρό μου είναι η αναγνώριση*
Μου αρέσουν οι έντονες συζητήσεις	*Αποφεύγω τις διαφωνίες*
J	***P***
Είμαι σοβαρός και συμβατικός	*Είμαι εύθυμος και αντισυμβατικός*
Είμαι πάντα στην ώρα μου	*Συνήθως αργώ*
Πρώτα η δουλειά και μετά η διασκέδαση	*Πρώτα η διασκέδαση και μετά η δουλειά*
Σέβομαι τους κανόνες	*Αμφισβητώ τη χρησιμότητα πολλών κανόνων*
Μου αρέσει να προγραμματίζω	*Θέλω να είμαι ευέλικτος*

E.3 Key concepts of the Games used

Accordingly, to the creators of these gaming activities, the programming key concepts taught by the course are (Code.org®, 2014):

- Programming Principles
    - Programming concepts Sequencing
    - Loops
    - Conditionals
    - Functions
    - Functions with parameters
    - Variables
- Computational Thinking
    - Decomposition
    - Patterns
    - Abstraction
    - Algorithm

Brief explanation of the blocks used to make commands and a description of what each command does of “K-8 Intro to CS” puzzles of Code.org®.

	The move forward block advances a character one space on the grid of the maze.
	The turn left block makes the character face to the left. It does not advance the character any spaces on the grid
	The turn right block makes the character face to the right. It does not advance the character any spaces on the grid.
	The “repeat times” block is called a loop in computer science. Whatever blocks you put inside this block will be repeated however many times you write in the top of the block. For example, if you put a “move forward” block followed by a “turn left” block inside the repeat times block and set it to repeat for 3 times, this is what will run: move forward, turn left, move forward, turn left, move forward, turn left.
	The “repeat until” block is also a loop. It will repeat (or loop) whatever command you put inside of it until it reaches the end goal or runs into a wall in the maze. This block is useful in cases where you don’t know exactly how many times you need something to repeat until a specific end point.
	The “if” block makes a decision for the computer. It checks to see if the statement on the top is true, and if it is true, the character will do whatever commands you put inside the block. If the statement on the top is not true, the character will do nothing. Note: when you click on the words “to the left”, it reveals a drop-down menu where you can choose other statements, the character can check: if there’s a path to the right or if there’s a path ahead.
	The “if else” block is very similar to the “if” block, but instead of doing nothing if the statement on the top is true, you can have it do an action or set of actions by placing commands in the “else” spot. For example, if you put a “move forward” block in the spot where it says “do” and you put a “turn left” block where it says “else”, the character will check to see if there’s a path ahead, and it there is, it will move forward, and if there’s not a path ahead, it will turn left. Note: there is also a drop-down menu in this block, just like the if block.

Bibliography

[1] Century, J., et al., *Building an operating system for computer science.* Chicago, IL: CEMSE, University of Chicago with UEI, University of Chicago. Retrieved November, 2013. 24: p. 2015.

[2] Address, T.S.o.t.U. 2016; Available from: https://www.whitehouse.gov/sotu. (Retreived online 10/06/2016)

[3] Smith, M. *Computer Science For All*. 2016; Available from: https://www.whitehouse.gov/blog/2016/01/30/computer-science-all. (Retreived online 10/06/2016)

[4] *K-12 Computer Science Framework*. 2016; Available from: http://www.k12cs.org/. (03/11/2016)

[5] Hubwieser, P., et al. *Towards a conceptualization of pedagogical content knowledge for computer science*. in *Proceedings of the ninth annual international ACM conference on International computing education research*. 2013. ACM.

[6] D'Ambrosio, U., *The role of mathematics in educational systems.* ZDM, 2007. 39(1-2): p. 173-181.

[7] Stephenson, C., et al., *The new educational imperative: Improving high school computer science education.* Computer Science Teachers Association (CSTA), New York, New York, 2005.

[8] Cooper, S., et al., *The Importance of Computing Education Research.* arXiv preprint arXiv:1604.03446, 2016.

[9] Wilson, C., et al., *Running on empty: The failure to teach K-12 computer science in the digital age.* Association for Computing Machinery, 2010.

[10] Committee, A.K.-T.F.C., *A model curriculum for K-12 computer science*. 2004, ACM Press.

[11] Tucker, A., et al., *A model curriculum for K-12 computer science.* Final Report of the ACM K-12 Task Force Curriculum Committee, CSTA, 2003.

[12] Hazzan, O., Gal-Ezer, J., and Blum, L. *A model for high school computer science education: the four key elements that make it!* in *ACM SIGCSE Bulletin*. 2008. ACM.

[13] Hazzan, O. and Lapidot, T., *The practicum in computer science education: bridging gaps between theoretical knowledge and actual performance.* ACM SIGCSE Bulletin, 2004. 36(4): p. 47-51.

[14] Hazzan, O. and Ragonis, N. *STEM teaching as an additional profession for scientists and engineers: the case of computer science education*. in *Proceedings of the 45th ACM technical symposium on Computer science education*. 2014. ACM.

[15] Zweben, S. and Bizot, B., *CRA Taulbee Survey Report 2014: Relentless Growth in Undergraduate CS Enrollment; Doctoral Degree Production Remains Strong but No New Record.* Computing Research News, 2015.

[16] Furber, S., *Shut down or restart? The way forward for computing in UK schools.* The Royal Society, London, 2012.

[17] Caspersen, M.E. and Nowack, P. *Computational thinking and practice: A generic approach to computing in Danish high schools*. in *Proceedings of the Fifteenth Australasian Computing Education Conference-Volume 136*. 2013. Australian Computer Society, Inc.

[18] Bell, T., Andreae, P., and Lambert, L. *Computer science in New Zealand high schools*. in *Proceedings of the Twelfth Australasian Conference on Computing Education-Volume 103*. 2010. Australian Computer Society, Inc.

[19] Bell, T., Andreae, P., and Robins, A., *A case study of the introduction of computer science in NZ schools.* ACM Transactions on Computing Education (TOCE), 2014. 14(2): p. 10.

[20] Gal-Ezer, J. and Stephenson, C., *Computer science teacher preparation is critical.* ACM Inroads, 2010. 1(1): p. 61-66.

[21] Lye, S.Y. and Koh, J.H.L., *Review on teaching and learning of computational thinking through programming: What is next for K-12?* Computers in Human Behavior, 2014. 41: p. 51-61.

[22] Ni, L. and Guzdial, M. *Who AM I?: understanding high school computer science teachers' professional identity*. in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 2012. ACM.

[23] Yadav, A., et al., *Expanding computer science education in schools: understanding teacher experiences and challenges.* Computer Science Education, 2016: p. 1-20.

[24] Cuny, J., *Transforming high school computing: a call to action.* ACM Inroads, 2012. 3(2): p. 32-36.

[25] Yadav, A., Hong, H., and Stephenson, C., *Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms.* TechTrends, 2016: p. 1-4.

[26] Stes, A., Gijbels, D., and Van Petegem, P., *Student-focused approaches to teaching in relation to context and teacher characteristics.* Higher Education, 2008. 55(3): p. 255-267.

[27] Kane, R., Sandretto, S., and Heath, C., *Telling half the story: A critical review of research on the teaching beliefs and practices of university academics.* Review of educational research, 2002. 72(2): p. 177-228.

[28] Ho, A., Watkins, D., and Kelly, M., *The conceptual change approach to improving teaching and learning: An evaluation of a Hong Kong staff development programme.* Higher Education, 2001. 42(2): p. 143-169.

[29] Stes, A., Coertjens, L., and Van Petegem, P., *Instructional development for teachers in higher education: impact on teaching approach.* Higher education, 2010. 60(2): p. 187-204.

[30] Danielson, C., *Enhancing professional practice: A framework for teaching*. 2011: ASCD.

[31] Quinn, H., Schweingruber, H., and Keller, T., *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. 2012: National Academies Press.

[32] Education., I.S.f.T.i. *2008 ISTE Standards for Teachers*. 2008; Available from: https://www.iste.org/resources/product?id=480. (10/11/2016)

[33] Holmboe, C., McIver, L., and George, C. *Research agenda for computer science education*. in *13th Workshop of the Psychology of Programming Interest Group*. 2001.

[34] Education, E.D.f. *National curriculum in England: Computing programmes of study*. 2013; Available from: https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study. (10/11/2016)

[35] Hubwieser, P. *The darmstadt model: a first step towards a research framework for computer science education in schools*. in *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. 2013. Springer.

[36] Sysło, M.M. and Kwiatkowska, A.B. *Introducing a new computer science curriculum for all school levels in Poland*. in *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. 2015. Springer.

[37] Board., C. *AP Program Participation and Performance Data 2015*. 2015; Available from: https://research.collegeboard.org/programs/ap/data/archived/ap-2015. (10/11/2016)

[38] Joint Task Force on Computing Curricula, A.f.C.M. and Society, I.C., *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. 2013: ACM. 518.

[39] Hubwieser, P., *The darmstadt model: a first step towards a research framework for computer science education in schools*, in *Proceedings of the 6th international conference on Informatics in Schools: Situation, Evolution, and Perspectives*. 2013, Springer-Verlag: Oldenburg, Germany. p. 1-14.

[40] Porter, L. and Simon, B. *Retaining nearly one-third more majors with a trio of instructional best practices in CS1*. in *Proceeding of the 44th ACM technical symposium on Computer science education*. 2013. ACM.

[41] Kaufman, W., *How one college is closing the computer science gender gap.* National Public Radio, May, 2013. 3.

[42] Theodoropoulos, A., Antoniou, A., and Lepouras, G., *Students teach students: Alternative teaching in Greek secondary education.* Education and Information Technologies, 2016. 21(2): p. 373-399.

[43] Θεοδώρου, Α., Μπέλλου, Ι., and Μικρόπουλος, Α., *Η έρευνα στη Διδακτική της Πληροφορικής στην Ελλάδα: Μία θεώρηση δεκαετίας.* Θέματα Επιστημών και Τεχνολογίας στην Εκπαίδευση, 2015. 7(3): p. 181-197.

[44] Kelleher, C. and Pausch, R., *Using storytelling to motivate programming.* Communications of the ACM, 2007. 50(7): p. 58-64.

[45] Biglan, A., *The characteristics of subject matter in different academic areas.* Journal of applied Psychology, 1973. 57(3): p. 195.

[46] Becher, T., *The significance of disciplinary differences.* Studies in Higher education, 1994. 19(2): p. 151-161.

[47] Trowler, P.R., *Academic tribes and territories*. 2001: McGraw-Hill Education (UK).

[48] Tucker, A., *A Model Curriculum for K--12 Computer Science: Final Report of the ACM K--12 Task Force Curriculum Committee.* 2003.

[49] Kirschner, P. and Wopereis, I.G., *Mindtools for teacher communities: A European perspective.* Technology, Pedagogy and Education, 2003. 12(1): p. 105-124.

[50] Shackelford, R., et al. *Computing curricula 2005: The overview report*. in *ACM SIGCSE Bulletin*. 2006. ACM.

[51] Curricula, C., *Report of the ACM/IEEE-CS Joint Curriculum Task Force*. 1991, IEEE Computer Society Press [ISBN 0-8186-2220-2] and ACM Press [ISBN 0-8979-381-7].

[52] Merritt, S.M., et al., *ACM model high school computer science curriculum.* Commun. ACM, 1993. 36(5): p. 87-90.

[53] Curricula, C., *Association for Computing Machinery and Computer Society of IEEE.* 2001.

[54] Lunt, B.M., et al., *Curriculum guidelines for undergraduate degree programs in information technology.* Retrieved March, 2008. 2: p. 2009.

[55] LeBlanc, R.J., et al., *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. 2006: IEEE Computer Society.

[56] CSTA, K., *Computer science standards.* Computer Science Teachers Association.

[57] Hawkridge, D., Boyd‐Barrett, O., and Scanlon, E., *Software for schools: British reviews in the late 1980s.* Computers and learning, 1990: p. 88-108.

[58] Oettinger, A.G. and Marks, S., *Run, Computer, Run; The Mythology of Educational Innovation.* 1969.

[59] Nager, A.A., Robert D. . *The Case for Improving U.S. Computer Science Education*. 2016; Available from: http://www2.itif.org/2016-computer-science-education.pdf?_ga=1.75349015.423871943.1479996292. (20/12/2016)

[60] Henderson, P.B., Cortina, T.J., and Wing, J.M. *Computational thinking*. in *ACM SIGCSE Bulletin*. 2007. ACM.

[61] Wilson, C. and Guzdial, M., *How to make progress in computing education.* Communications of the ACM, 2010. 53(5): p. 35-37.

[62] Committee, A.T.-Y.C.E., *Computing Curricula 2009: Guidelines for Associate-Degree Transfer Curriculum in Computer Science*. 2008.

[63] Cassel, L.N., et al., *The computing ontology: application in education*, in *Working group reports on ITiCSE on Innovation and technology in computer science education*. 2007, ACM: Dundee, Scotland. p. 171-183.

[64] Schwill, A., *Fundamental ideas of computer science.* Bulletin-European Association for Theoretical Computer Science, 1994. 53: p. 274-274.

[65] Ershov, A., *Programming, the second literacy.* Microprocessing and Microprogramming, 1981. 8(1): p. 1-9.

[66] Guzdial, M., *Learner-Centered Design of Computing Education: Research on Computing for Everyone.* Synthesis Lectures on Human-Centered Informatics, 2015. 8(6): p. 1-165.

[67] Wing, J.M., *Computational thinking.* Communications of the ACM, 2006. 49(3): p. 33-35.

[68] Rotherham, A.J. and Willingham, D.T., *"21st-Century" Skills.* American Educator, 2010: p. 17.

[69] Hromkovič, J. *Contributing to general education by teaching informatics*. in *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*. 2006. Springer.

[70] Ragonis, N., *Computing Pre‐University: Secondary Level Computing Curricula.* Wiley encyclopedia of computer science and engineering, 2008.

[71] Sahami, M., et al. *Acm/ieee-cs computer science curriculum 2013: Reviewing the ironman report*. in *Proceeding of the 44th ACM technical symposium on Computer science education*. 2013. ACM.

[72] Κόμης, Β., *Εισαγωγή στη Διδακτική της Πληροφορικής.* Εκδόσεις Κλειδάριθμος, Αθήνα, 2005.

[73] Parkay, F.W., Stanford, B.H., and Gougeon, T.D., *Becoming a teacher*. 2010: Pearson/Merrill.

[74] Grimmett, P.P. and MacKinnon, A.M., *Craft knowledge and the education of teachers.* Review of research in education, 1992. 18: p. 385-456.

[75] Shulman, L.S., *Those who understand: Knowledge growth in teaching.* Educational researcher, 1986. 15(2): p. 4-14.

[76] Shulman, L., *Knowledge and teaching: Foundations of the new reform.* Harvard educational review, 1987. 57(1): p. 1-23.

[77] Gess-Newsome, J., *Pedagogical content knowledge: An introduction and orientation*, in *Examining pedagogical content knowledge*. 1999, Springer. p. 3-17.

[78] Hill, H.C., Ball, D.L., and Schilling, S.G., *Unpacking pedagogical content knowledge: Conceptualizing and measuring teachers' topic-specific knowledge of students.* Journal for research in mathematics education, 2008: p. 372-400.

[79] Powell, D.R., et al., *Effects of an early literacy professional development intervention on head start teachers and children.* Journal of Educational Psychology, 2010. 102(2): p. 299.

[80] Saeli, M., et al., *Teaching programming in secondary school: a pedagogical content knowledge perspective.* Informatics in Education-An International Journal, 2011(Vol 10_1): p. 73-88.

[81] Liberman, N., Kolikant, Y.B.-D., and Beeri, C., *"Regressed experts" as a new state in teachers' professional development: lessons from Computer Science teachers' adjustments to substantial changes in the curriculum.* Computer Science Education, 2012. 22(3): p. 257-283.

[82] Veenman, S., *Perceived problems of beginning teachers.* Review of educational research, 1984. 54(2): p. 143-178.

[83] Lortie, D.C. and Clement, D., *Schoolteacher: A sociological study*. 1975: JSTOR.

[84] Feiman-Nemser, S., et al., *A Conceptual Review of Literature on New Teacher Induction.* 1999.

[85] Huling, L., Resta, V., and Yeargain, P., *Supporting and retaining novice teachers.* Kappa Delta Pi Record, 2012. 48(3): p. 140-143.

[86] Martin, P., *Novice Teachers: Meeting the Challenge.* Principal, 2008. 88(2): p. 42-44.

[87] Shuell, T.J., *Cognitive conceptions of learning.* Review of educational research, 1986. 56(4): p. 411-436.

[88] Wadsworth, B.J., *Piaget's theory of cognitive and affective development: Foundations of constructivism*. 1996: Longman Publishing.

[89] Bruner, J.S., *The process of education*. 2009: Harvard University Press.

[90] Vygotskiĭ, L.S., Hanfmann, E., and Vakar, G., *Thought and language*. 2012: MIT press.

[91] Skinner, B., *The Behavior of Organisms (Appleton-Century-Crofts, New York).* Spaulding, WD, Storms, L., Goodrich, V., and Sullivan, M.(1986), Applications of experimental psychopathology in psychiatric rehabilitation. Schizophrenia Bulletin, 1938. 12: p. 560577.

[92] Skinner, B.F., *Teaching science in high school-What is wrong.* Science, 1968. 159(3816): p. 704-710.

[93] Schunk, D.H., *Learning theories.* Printice Hall Inc., New Jersey, 1996.

[94] Brown, A.L. and Palincsar, A.S., *Guided, cooperative learning and individual knowledge acquisition.* Knowing, learning, and instruction: Essays in honor of Robert Glaser, 1989: p. 393-451.

[95] Brown, A.L., Palincsar, A.S., and Armbruster, B.B., *Instructing comprehension-fostering activities in interactive learning situations.* Learning and comprehension of text, 1984: p. 255-286.

[96] Wood, D., *How children think and learn: The social contexts of cognitive development*. 1998: Blackwell Publishing.

[97] Ben-Ari, M., *Constructivism in computer science education.* Journal of Computers in Mathematics and Science Teaching, 2001. 20(1): p. 45-74.

[98] Papert, S., *Mindstorms: Children, computers, and powerful ideas*. 1980: Basic Books, Inc.

[99] Piaget, J., *The construction of reality in the child*. Vol. 82. 2013: Routledge.

[100] Ben-Ari, M. *Constructivism in computer science education*. in *Acm sigcse bulletin*. 1998. ACM.

[101] Quevedo-Torrero, J.U. *Learning theories in computer science education*. in *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*. 2009. IEEE.

[102] Lave, J. and Wenger, E., *Situated learning: Legitimate peripheral participation.* 1991: Cambridge university press.

[103] Wenger, E., *Communities of practice: Learning, meaning, and identity*. 1998: Cambridge university press.

[104] Kolikant, Y.B.-D., *Learning concurrency as an entry point to the community of computer science practitioners.* JOURNAL OF COMPUTERS IN MATHEMATICS AND SCIENCE TEACHING., 2004. 23(1): p. 21-46.

[105] Bloom, B.S., *Taxonomy of educational objectives: The classification of educational goals.* 1956.

[106] Anderson, L.W., Krathwohl, D.R., and Bloom, B.S., *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. 2001: Allyn & Bacon.

[107] Lister, R. and Leaney, J., *Introductory programming, criterion-referencing, and bloom.* ACM SIGCSE Bulletin, 2003. 35(1): p. 143-147.

[108] Thompson, E., et al. *Bloom's taxonomy for CS assessment*. in *Proceedings of the tenth conference on Australasian computing education-Volume 78*. 2008. Australian Computer Society, Inc.

[109] Johnson, C.G. and Fuller, U. *Is Bloom's taxonomy appropriate for computer science?* in *Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006*. 2006. ACM.

[110] Kolb, D., *Experiential learning as the science of learning and development.* Englewood Cliffs NPH, editor1984, 1984.

[111] Perry Jr, W.G., *Forms of Intellectual and Ethical Development in the College Years: A Scheme. Jossey-Bass Higher and Adult Education Series*. 1999: ERIC.

[112] Perry, W. and Chickering, A., *Cognitive and ethical growth: The making of meaning.* College student development and academic life, 1997. 4: p. 48-116.

[113] Entwistle, N. and Tait, H., *Approaches to studying and perceptions of the learning environment across disciplines.* New directions for teaching and learning, 1995. 1995(64): p. 93-103.

[114] Myers, I.B., et al., *MBTI manual: A guide to the development and use of the Myers-Briggs Type Indicator*. Vol. 3. 1998: Consulting Psychologists Press Palo Alto, CA.

[115] McLoughlin, C., *The implications of the research literature on learning styles for the design of instructional material.* Australasian Journal of Educational Technology, 1999. 15(3): p. 222-241.

[116] Curry, L., *Integrating concepts of cognitive or learning style: A review with attention to psychometric standards*. 1987: Canadian College of Health Service Executives.

[117] Dunn, R.S. and Dunn, K.J., *Teaching students through their individual learning styles: A practical approach*. 1978: Prentice Hall.

[118] Riding, R. and Cheema, I., *Cognitive styles—an overview and integration.* Educational psychology, 1991. 11(3-4): p. 193-215.

[119] Riding, R. and Rayner, S., *Cognitive styles and learning strategies: Understanding style differences in learning and behavior*. 2013: Routledge.

[120] Prince, M.J. and Felder, R.M., *Inductive teaching and learning methods: Definitions, comparisons, and research bases.* Journal of engineering education, 2006. 95(2): p. 123-138.

[121] Jonassen, D.H., *Objectivism versus constructivism: Do we need a new philosophical paradigm?* Educational technology research and development, 1991. 39(3): p. 5-14.

[122] Saroyan, A., *The lecturer: Working with large groups.* Teaching Alone, Teaching Together, Jossey-Bass, 2000: p. 87-107.

[123] Charlton, B.G., *Lectures are such an effective teaching method because they exploit evolved human psychology to improve learning.* Medical hypotheses, 2006. 67(6): p. 1261-1265.

[124] Prince, M., *Does active learning work? A review of the research.* Journal of engineering education, 2004. 93(3): p. 223-231.

[125] Faust, J.L. and Paulson, D.R., *Active learning in the college classroom.* Journal on Excellence in College Teaching, 1998. 9(2): p. 3-24.

[126] Kolb, D.A., *Learning styles inventory.* The Power of the 2 2 Matrix, 2000: p. 267.

[127] Fleming, N.D. and Mills, C., *Not another inventory, rather a catalyst for reflection.* 1992.

[128] Fleming, N. and Baume, D., *Learning Styles Again: VARKing up the right tree!* Educational Developments, 2006. 7(4): p. 4.

[129] Felder, R.M. and Silverman, L.K., *Learning and teaching styles in engineering education.* Engineering education, 1988. 78(7): p. 674-681.

[130] Bogdan, R. and Biklen, S.K., *Qualitative research for education.* 1997: Allyn & Bacon Boston.

[131] Freeman, S., et al., *Active learning increases student performance in science, engineering, and mathematics.* Proceedings of the National Academy of Sciences, 2014. 111(23): p. 8410-8415.

[132] Limbach, B.J. and Waugh, W.L., *Questioning the lecture format.* The NEA Higher Education Journal: Thought and Action, 2005. 20(1): p. 47-56.

[133] Daly, J.A., et al., *Pre‐performance concerns associated with public speaking anxiety.* Communication Quarterly, 1989. 37(1): p. 39-53.

[134] Lucas, S.E., *The Art of Public Speaking 11th Edition*. 2008: McGraw-Hill.

[135] Thomas, E., *The variation of memory with time for information appearing during a lecture.* Studies in adult education, 1972. 4(1): p. 57-62.

[136] Chickering, A.W. and Gamson, Z.F., *Seven principles for good practice in undergraduate education.* AAHE bulletin, 1987. 3: p. 7.

[137] Meyers, C. and Jones, T.B., *Promoting Active Learning. Strategies for the College Classroom*. 1993: ERIC.

[138] McParland, M., Noble, L.M., and Livingston, G., *The effectiveness of problem‐based learning compared to traditional teaching in undergraduate psychiatry*. Medical education, 2004. 38(8): p. 859-867.

[139] Morrison, G.R., et al., *Designing effective instruction*. 2010: John Wiley & Sons.

[140] Topping, K.J., *Trends in peer learning*. Educational psychology, 2005. 25(6): p. 631-645.

[141] Boud, D., *Introduction: Making the move to peer learning*. Peer learning in higher education, 2001: p. 1-17.

[142] Caldwell, J.E., *Clickers in the large classroom: Current research and best-practice tips*. CBE-Life sciences education, 2007. 6(1): p. 9-20.

[143] Lopez, M., et al. *Relationships between reading, tracing and writing skills in introductory programming*. in *Proceedings of the fourth international workshop on computing education research*. 2008. ACM.

[144] Keppell, M., et al., *Peer learning and learning‐oriented assessment in technology‐enhanced environments*. Assessment & Evaluation in Higher Education, 2006. 31(4): p. 453-464.

[145] Kothiyal, A., et al. *Effect of think-pair-share in a large CS1 class: 83% sustained engagement*. in *Proceedings of the ninth annual international ACM conference on International computing education research*. 2013. ACM.

[146] Porter, L., Bailey Lee, C., and Simon, B. *Halving fail rates using peer instruction: a study of four computer science courses*. in *Proceeding of the 44th ACM technical symposium on Computer science education*. 2013. ACM.

[147] Simon, B., et al. *Experience report: peer instruction in introductory computing*. in *Proceedings of the 41st ACM technical symposium on Computer science education*. 2010. ACM.

[148] Simon, B., Parris, J., and Spacco, J. *How we teach impacts student learning: peer instruction vs. lecture in CS0*. in *Proceeding of the 44th ACM technical symposium on Computer science education*. 2013. ACM.

[149] Zingaro, D. *Peer instruction contributes to self-efficacy in CS1*. in *Proceedings of the 45th ACM technical symposium on Computer science education*. 2014. ACM.

[150] Wills, C.E., et al., *Peer learning in an introductory computer science course*. ACM SIGCSE Bulletin, 1994. 26(1): p. 309-313.

[151] Griffiths, S., et al., *Enhancing Student Learning Through Peer Tutoring in Higher Education: A Compendium Resource Pack*. 1995: University of Ulster.

[152] Goodlad, S. and Hirst, B., *Peer Tutoring. A Guide to Learning by Teaching*. 1989: ERIC.

[153] Brookfield, S.D. and Preskill, S., *Discussion as a way of teaching*. Vol. 85. 1999: Citeseer.

[154] Johnson, D.W., Johnson, R.T., and Smith, K.A., *Active learning: Cooperation in the college classroom*. 1998: ERIC.

[155] Slavin, R.E., *Cooperative learning: Theory, research, and practice*. 1990: Englewood Cliffs, NJ: Prentice-Hall.

[156] Tannenberg, J. *Using cooperative learning in the undergraduate computer science classroom*. in *Proceedings of the Midwest Small College Computing Conference*. 1995.

[157] Mazur, E., *Peer instruction*. 1997: Upper Saddle River, NJ: Prentice Hall.

[158] Sadler, P.M. and Good, E., *The impact of self-and peer-grading on student learning*. Educational assessment, 2006. 11(1): p. 1-31.

[159] Barrows, H.S. and Tamblyn, R.M., *Problem-based learning: An approach to medical education*. 1980: Springer Publishing Company.

[160] Wood, D.F., *Problem based learning*. British medical journal, 2003. 326(7384): p. 328.

[161] Duch, B.J., Groh, S.E., and Allen, D.E., *The power of problem-based learning: a practical" how to" for teaching undergraduate courses in any discipline*. 2001: Stylus Publishing, LLC.

[162] Savery, J.R. and Duffy, T.M., *Problem based learning: An instructional model and its constructivist framework*. Educational technology, 1995. 35(5): p. 31-38.

[163] Boud, D. and Feletti, G., *The challenge of problem-based learning*. 1997: Psychology Press.

[164] Greening, T., et al. *Trialing a Problem-Based Learning Approach to First Year Computer Science*. in *Proceedings of the First Australasian Conference on Computer Science Education*. 1997. Citeseer.

[165] Savery, J.R., *Overview of problem-based learning: Definitions and distinctions*. Essential readings in problem-based learning: Exploring and extending the legacy of Howard S. Barrows, 2015: p. 5-15.

[166] Barrett, T., *The problem‐based learning process as finding and being in flow*. Innovations in Education and Teaching International, 2010. 47(2): p. 165-174.

[167] Schmidt, H.G., *Problem‐based learning: Rationale and description*. Medical education, 1983. 17(1): p. 11-16.

[168] O'Kelly, J. and Gibson, J.P. *RoboCode & problem-based learning: a non-prescriptive approach to teaching programming*. in *ACM SIGCSE Bulletin*. 2006. ACM.

[169] Duch, B.J., *Writing problems for deeper understanding*. The power of problem-based learning, 2001: p. 47-58.

[170] Blumenfeld, P.C., et al., *Motivating project-based learning: Sustaining the doing, supporting the learning*. Educational psychologist, 1991. 26(3-4): p. 369-398.

[171] Krajcik, J.S., et al., *A collaborative model for helping middle grade science teachers learn project-based instruction*. The elementary school journal, 1994: p. 483-497.

[172] Marx, R.W., et al., *Inquiry‐based science in the middle grades: Assessment of learning in urban systemic reform*. Journal of research in Science Teaching, 2004. 41(10): p. 1063-1080.

[173] Rivet, A.E. and Krajcik, J.S., *Achieving standards in urban systemic reform: An example of a sixth grade project‐based science curriculum*. Journal of Research in Science Teaching, 2004. 41(7): p. 669-692.

[174] Bender, W.N., *Project-based learning: Differentiating instruction for the 21st century*. 2012: Corwin Press.

[175] Savery, J.R., *Overview of problem-based learning: Definitions and distinctions*. Interdisciplinary Journal of Problem-based Learning, 2006. 1(1): p. 3.

[176] Bell, S., *Project-based learning for the 21st century: Skills for the future*. The Clearing House, 2010. 83(2): p. 39-43.

[177] Larmer, J., *Project-Based Learning vs. Problem-Based Learning vs. X-BL*. Retrieved March, 2014. 8: p. 2024.

[178] Perrenet, J., Bouhuijs, P., and Smits, J., *The suitability of problem-based learning for engineering education: theory and practice*. Teaching in higher education, 2000. 5(3): p. 345-358.

[179] Hundhausen, C.D., Narayanan, N.H., and Crosby, M.E., *Exploring studio-based instructional models for computing education*. ACM SIGCSE Bulletin, 2008. 40(1): p. 392-396.

[180] Green, L.N. and Bonollo, E., *Studio-based teaching: history and advantages in the teaching of design*. World Transactions on Eng. and Tech. Edu, 2003. 2(2): p. 269-272.

[181] Schön, D.A., *The reflective practitioner: How professionals think in action*. Vol. 5126. 1983: Basic books.

[182] Boyer, E.L. and Mitgang, L.D., *Building Community: A New Future for Architecture Education and Practice. A Special Report*. 1996: ERIC.

[183] Myneni, L., et al. *Studio-based learning in CS2: An experience report*. in *Proceedings of the 46th Annual Southeast Regional Conference on XX*. 2008. ACM.

[184] Carter, A.S. and Hundhausen, C.D., *A review of studio-based learning in computer science*. Journal of Computing Sciences in Colleges, 2011. 27(1): p. 105-111.

[185] Hübscher-Younger, T. and Narayanan, N.H., *Authority and convergence in collaborative learning*. Computers & education, 2003. 41(4): p. 313-334.

[186] Hundhausen, C.D., *Integrating algorithm visualization technology into an undergraduate algorithms course: ethnographic studies of a social constructivist approach*. Computers & Education, 2002. 39(3): p. 237-260.

[187] Hendrix, D., et al. *Implementing studio-based learning in CS2*. in *Proceedings of the 41st ACM technical symposium on Computer science education*. 2010. ACM.

[188] Docherty, M., et al. *An innovative design and studio-based CS degree*. in *ACM SIGCSE Bulletin*. 2001. ACM.

[189] Narayanan, N.H., et al. *Transforming the CS classroom with studio-based learning*. in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 2012. ACM.

[190] Hundhausen, C.D., Agrawal, A., and Agarwal, P., *Talking about code: Integrating pedagogical code reviews into early computing courses*. Trans. Comput. Educ., 2013. 13(3): p. 1-28.

[191] Grissom, S., Hundhausen, C., and Conrad, P., *Alternatives to lecture: experience peer instruction and pedagogical code reviews*, in *Proceedings of the 45th ACM technical symposium on Computer science education*. 2014, ACM: Atlanta, Georgia, USA. p. 275-276.

[192] Barrett, T., *Understanding problem-based learning*. Handbook of enquiry and problem-based learning, 2005: p. 13-25.

[193] Olson, S. and Loucks-Horsley, S., *Inquiry and the National Science Education Standards: A guide for teaching and learning*. 2000: National Academies Press.

[194] Bruner, J.S., *The act of discovery*. Harvard educational review, 1961.

[195] Edelson, D.C., Gordin, D.N., and Pea, R.D., *Addressing the challenges of inquiry-based learning through technology and curriculum design*. Journal of the learning sciences, 1999. 8(3-4): p. 391-450.

[196] Fay, M.E., et al., *A rubric to characterize inquiry in the undergraduate chemistry laboratory*. Chemistry Education Research and Practice, 2007. 8(2): p. 212-219.

[197] Lawson, A.E., *A Theory of Instruction: Using the Learning Cycle To Teach Science Concepts and Thinking Skills. NARST Monograph, Number One, 1989*. 1989.

[198] Atkin, J.M. and Karplus, R., *Discovery or invention?* The Science Teacher, 1962. 29(5): p. 45-51.

[199] Abernethy, K., Gabbert, P., and Treu, K. *Inquiry-based computer science instruction: Some initial experiences*. in *ACM SIGCSE Bulletin*. 1998. ACM.

[200] Hmelo-Silver, C.E., Duncan, R.G., and Chinn, C.A., *Scaffolding and achievement in problem-based and inquiry learning: A response to Kirschner, Sweller, and Clark (2006)*. Educational psychologist, 2007. 42(2): p. 99-107.

[201] Eberlein, T., et al., *Pedagogies of engagement in science*. Biochemistry and molecular biology education, 2008. 36(4): p. 262-273.

[202] Moog, R.S., et al. *POGIL: Process oriented guided inquiry learning*. in *ACS Symposium series*. 2008.

[203] Brown, P.J., *Process-oriented guided-inquiry learning in an introductory anatomy and physiology course with a diverse student population*. Advances in physiology education, 2010. 34(3): p. 150-155.

[204] Hu, H.H. and Shepherd, T.D., *Using POGIL to help students learn to program*. Trans. Comput. Educ., 2013. 13(3): p. 1-23.

[205] Kussmaul, C. *Process oriented guided inquiry learning for soft computing*. in *International Conference on Advances in Computing and Communications*. 2011. Springer.

[206] Hu, H.H. and Kussmaul, C. *Promoting student-centered learning with POGIL*. in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 2012. ACM.

[207] Hu, H.H. and Shepherd, T.D. *Teaching CS 1 with POGIL activities and roles*. in *Proceedings of the 45th ACM technical symposium on Computer science education*. 2014. ACM.

[208] Monge, A.E., et al., *EngageCSEdu: engaging and retaining CS1 and CS2 students.* ACM Inroads, 2015. 6(1): p. 6-11.

[209] Kussmaul, C. *Process oriented guided inquiry learning (POGIL) for computer science*. in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 2012. ACM.

[210] Moog, R.S. and Spencer, J.N., *Process oriented guided inquiry learning (POGIL).* 2008: ACS Publications.

[211] Kussmaul, C., Hu, H., and Lang, M. *Using POGIL to help students discover CS concepts and develop process skills*. in *Proceeding of the 44th ACM technical symposium on Computer science education*. 2013. ACM.

[212] De Gale, S. and Boisselle, L.N., *The effect of POGIL on academic performance and academic confidence.* 2015.

[213] Olivieri, L.M., *Piloting POGIL in an introductory Python programming course.* Journal of Computing Sciences in Colleges, 2013. 28(6): p. 194-195.

[214] Kussmaul, C., Hu, H.H., and Pirmann, T. *Special Session: Helping Students to Develop Communication, Teamwork, and Other Process Skills with POGIL*. in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 2016. ACM.

[215] Hu, H.H., et al. *Results from a Survey of Faculty Adoption of Process Oriented Guided Inquiry Learning (POGIL) in Computer Science*. in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. 2016. ACM.

[216] Hu, H. and Avery, B., *CS principles with POGIL activities as a learning community.* Journal of Computing Sciences in Colleges, 2015. 31(2): p. 79-86.

[217] Hu, H.H., Kussmaul, C., and Lang, M. *Using POGIL activities in computer science classes*. in *Proceeding of the 44th ACM technical symposium on Computer science education*. 2013. ACM.

[218] Hu, H.H., et al. *Guided inquiry learning in context: perspectives on POGIL in CS*. in *Proceedings of the 45th ACM technical symposium on Computer science education*. 2014. ACM.

[219] Vanderhyde, J. and Appel, F., *With greater CS enrollments comes an even greater need for engaging teaching practices.* Journal of Computing Sciences in Colleges, 2016. 32(1): p. 38-45.

[220] Hackman, J.R., *The design of work teams.* Psychological Dimensions of Organizational Behavior: 2d Edition. Ed. Barry M. Staw. New Jersey: Prentice Hall, Inc, 1995.

[221] Haring-Smith, T., *Learning together: An introduction to collaborative learning*. 1993: HarperCollins College Publ.

[222] Dillenbourg, P., et al., *The evolution of research on collaborative learning*. 1995, Elsevier, Oxford.

[223] Matthews, R.S., et al., *Building bridges between cooperative and collaborative learning.* Change: The magazine of higher learning, 1995. 27(4): p. 35-40.

[224] Crouch, C.H. and Mazur, E., *Peer instruction: Ten years of experience and results.* American journal of physics, 2001. 69(9): p. 970-977.

[225] Sangin, M., et al., *Facilitating peer knowledge modeling: Effects of a knowledge awareness tool on collaborative learning outcomes and processes.* Computers in Human Behavior, 2011. 27(3): p. 1059-1067.

[226] Kleiman, P., *Towards transformation: conceptions of creativity in higher education.* Innovations in Education and Teaching International, 2008. 45(3): p. 209-217.

[227] Requena-Carrión, J., et al. *A student-centered collaborative learning environment for developing communication skills in engineering education*. in *Education Engineering (EDUCON), 2010 IEEE*. 2010. IEEE.

[228] Williams, L. and Kessler, R., *Pair programming illuminated*. 2002: Addison-Wesley Longman Publishing Co., Inc.

[229] McDowell, C., et al., *Pair programming improves student retention, confidence, and program quality.* Communications of the ACM, 2006. 49(8): p. 90-95.

[230] McDowell, C., et al. *The impact of pair programming on student performance, perception and persistence*. in *Proceedings of the 25th international conference on Software engineering*. 2003. IEEE Computer Society.

[231] Jensen, R., *A pair programming experience.* CrossTalk, 2003. 16(3): p. 22-24.

[232] Beck, K., *Extreme programming explained: embrace change*. 2000: addison-wesley professional.

[233] Hanks, B., et al., *Pair programming in education: a literature review.* Computer Science Education, 2011. 21(2): p. 135-173.

[234] Chaparro, E.A., et al. *Factors affecting the perceived effectiveness of pair programming in higher education*. in *Proc. PPIG*. 2005.

[235] Williams, L., et al. *Building pair programming knowledge through a family of experiments*. in *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*. 2003. IEEE.

[236] Cockburn, A. and Williams, L., *The costs and benefits of pair programming.* Extreme programming examined, 2000: p. 223-247.

[237] Nagappan, N., et al., *Improving the CS1 experience with pair programming.* ACM SIGCSE Bulletin, 2003. 35(1): p. 359-362.

[238] Hanks, B., et al. *Program quality with pair programming in CS1*. in *ACM SIGCSE Bulletin*. 2004. ACM.

[239] Salleh, N., Mendes, E., and Grundy, J., *Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review.* IEEE Transactions on Software Engineering, 2011. 37(4): p. 509-525.

[240] Jones, J.S. *Participatory teaching methods in computer science*. in *ACM SIGCSE Bulletin*. 1987. ACM.

[241] Schuler, D. and Namioka, A., *Participatory design: Principles and practices*. 1993: CRC Press.

[242] Bjerknes, G., et al., *Computers and democracy: A Scandinavian challenge*. 1987: Gower Pub Co.

[243] Greenbaum, J. and Kyng, M., *Design by doing*. 1991: Lawrence Erlbaum Associates, Incorporated.

[244] Muller, M.J. and Kuhn, S., *Participatory design*. Communications of the ACM, 1993. 36(6): p. 24-28.

[245] Muller, M.J., *Participatory design: the third space in HCI*. Human-computer interaction: Development process, 2003. 4235: p. 165-185.

[246] Weinberg, J.B. and Stephen, M.L. *Participatory design in a human-computer interaction course: teaching ethnography methods to computer scientists*. in *ACM SIGCSE Bulletin*. 2002. ACM.

[247] Gasen, J.B., *HCI education: past, present and future?* ACM SIGCHI Bulletin, 1996. 28(1): p. 25-27.

[248] Kautz, K., *User participation and participatory design: Topics in computing education*. Human–Computer Interaction, 1996. 11(3): p. 267-284.

[249] Antoniou, A. and Lepouras, G. *A fast pace method for involving children in edutainment-technology design*. in *Advances in Computer-Human Interaction, 2008 First International Conference on*. 2008. IEEE.

[250] Hecht, K.M. and Maass, S. *Teaching participatory design*. in *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008*. 2008. Indiana University.

[251] Liddle, D. *Design of the conceptual model*. in *Bringing design to software*. 1996. ACM.

[252] Pivec, M., Dziabenko, O., and Schinnerl, I. *Aspects of game-based learning*. in *3rd International Conference on Knowledge Management, Graz, Austria*. 2003.

[253] Kirriemuir, J. and McFarlane, A., *Literature review in games and learning*. 2004.

[254] Marc, P., *Digital game-based learning*. 2001, McGraw-Hill: New York.

[255] Blackman, S., *Serious games... and less!* ACM Siggraph Computer Graphics, 2005. 39(1): p. 12-16.

[256] Prensky, M., *Digital game-based learning*. Computers in Entertainment (CIE), 2003. 1(1): p. 21-21.

[257] Thornton, G.C. and Cleveland, J.N., *Developing managerial talent through simulation*. American Psychologist, 1990. 45(2): p. 190.

[258] de Felix, J.W. and Johnson, R.T., *Learning from video games*. Computers in the Schools, 1993. 9(2-3): p. 119-134.

[259] Baranauskas, C.C., Neto, N.G.G., and Borges, M.A., *Learning at work through a multi-user synchronous simulation game*. International Journal of Continuing Engineering Education and Life Long Learning, 2001. 11(3): p. 251-260.

[260] Malone, T.W., *Toward a theory of intrinsically motivating instruction*. Cognitive science, 1981. 5(4): p. 333-369.

[261] Prensky, M., *Digital game-based learning*. 2001, McGraw-Hill: New York.

[262] Michael, D.R. and Chen, S.L., *Serious games: Games that educate, train, and inform*. 2005: Muska & Lipman/Premier-Trade.

[263] Williamson, B., *Computer games, schools, and young people: A report for educators on using games for learning*. 2009: Futurelab Bristol.

[264] Hong, J.C., et al., *Assessing the educational values of digital games*. Journal of Computer Assisted Learning, 2009. 25(5): p. 423-437.

[265] Pivec, M., *Editorial: Play and learn: potentials of game‐based learning*. British Journal of Educational Technology, 2007. 38(3): p. 387-393.

[266] De Freitas, S. and Oliver, M., *How can exploratory learning with games and simulations within the curriculum be most effectively evaluated?* Computers & Education, 2006. 46(3): p. 249-264.

[267] McFarlane, A., Sparrowhawk, A., and Heald, Y., *Report on the educational use of games*. 2002: TEEM (Teachers evaluating educational multimedia), Cambridge.

[268] Lepper, M.R. and Cordova, D.I., *A desire to be taught: Instructional consequences of intrinsic motivation*. Motivation and emotion, 1992. 16(3): p. 187-208.

[269] Connolly, T.M., et al., *A systematic literature review of empirical evidence on computer games and serious games*. Computers & Education, 2012. 59(2): p. 661-686.

[270] Papastergiou, M., *Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation*. Computers & Education, 2009. 52(1): p. 1-12.

[271] Gee, J.P., *What video games have to teach us about learning and literacy*. Computers in Entertainment (CIE), 2003. 1(1): p. 20-20.

[272] Charlier, N. and De Fraine, B. *Games-Based Learning in Teacher Education: A strategy to Integrate Digital Games into Secondary School*. in *2nd European Conference on Games Based Learning*. 2008.

[273] Hartness, K., *Robocode: using games to teach artificial intelligence*. Journal of Computing Sciences in Colleges, 2004. 19(4): p. 287-291.

[274] Esper, S., Foster, S.R., and Griswold, W.G. *CodeSpells: embodying the metaphor of wizardry for programming*. in *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*. 2013. ACM.

[275] Barnes, T., et al. *Game2Learn: building CS1 learning games for retention*. in *ACM SIGCSE Bulletin*. 2007. ACM.

[276] Chaffin, A., et al. *Experimental evaluation of teaching recursion in a video game*. in *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*. 2009. ACM.

[277] Muratet, M., et al. *Experimental feedback on Prog&Play: a serious game for programming practice*. in *Computer Graphics Forum*. 2011. Wiley Online Library.

[278] Lee, J.J. and Hammer, J., *Gamification in education: What, how, why bother?* Academic exchange quarterly, 2011. 15(2): p. 146.

[279] Hamari, J., *Transforming homo economicus into homo ludens: A field experiment on gamification in a utilitarian peer-to-peer trading service.* Electronic commerce research and applications, 2013. 12(4): p. 236-245.

[280] Hamari, J., Koivisto, J., and Sarsa, H. *Does gamification work?--a literature review of empirical studies on gamification*. in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. 2014. IEEE.

[281] Denny, P. *The effect of virtual achievements on student engagement*. in *Proceedings of the SIGCHI conference on human factors in computing systems*. 2013. ACM.

[282] Fitz-Walter, Z., Tjondronegoro, D., and Wyeth, P. *Orientation passport: using gamification to engage university students*. in *Proceedings of the 23rd Australian computer-human interaction conference*. 2011. ACM.

[283] Singer, L. and Schneider, K. *It was a bit of a race: Gamification of version control*. in *Games and Software Engineering (GAS), 2012 2nd International Workshop on*. 2012. IEEE.

[284] Simões, J., Redondo, R.D., and Vilas, A.F., *A social gamification framework for a K-6 learning platform*. Computers in Human Behavior, 2013. 29(2): p. 345-353.

[285] Li, W., Grossman, T., and Fitzmaurice, G. *GamiCAD: a gamified tutorial system for first time autocad users*. in *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 2012. ACM.

[286] DomíNguez, A., et al., *Gamifying learning experiences: Practical implications and outcomes.* Computers & Education, 2013. 63: p. 380-392.

[287] Sanchez, E., Young, S., and Jouneau-Sion, C., *Classcraft: from gamification to ludicization of classroom management.* Education and Information Technologies, 2016: p. 1-17.

[288] Caci, B., Cardaci, M., and Lund, H.H., *Assessing educational robotics by the "Robot edutainment questionnaire"*. Technical Report, 2003.

[289] Kara, D., *Sizing and seizing the robotics opportunity, RoboNexus.* Robotics Trends Inc, 2004.

[290] Catlin, D. and Robertson, S. *Using educational robots to enhance the performance of minority students*. in *TRTWR 2012 Conference, Riva La Garda Italy*. 2012.

[291] Johnson, J., *Children, robotics, and education.* Artificial Life and Robotics, 2003. 7(1-2): p. 16-21.

[292] Mikropoulos, T.A. and Bellou, I., *Educational robotics as mindtools.* Themes in Science and Technology Education, 2013. 6(1): p. 5-14.

[293] Rockland, R., et al., *Advancing the "E" in K-12 STEM education.* 2010.

[294] Alimisis, D., *Educational robotics: Open questions and new challenges.* Themes in Science and Technology Education, 2013. 6(1): p. 63-71.

[295] Sklar, E., Eguchi, A., and Johnson, J. *RoboCupJunior: learning with educational robotics*. in *Robot Soccer World Cup*. 2002. Springer.

[296] Theodoropoulos, A., Antoniou, A., and Lepouras, G. *Educational Robotics in the service of CSE: A study based on the PanHellenic competition*. in *Proceedings of the 11th Workshop in Primary and Secondary Computing Education*. 2016. ACM.

[297] Curto, B. and Moreno, V. *A robot in the classroom*. in *Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality*. 2013. ACM.

[298] Druin, A. and Hendler, J.A., *Robots for kids: exploring new technologies for learning*. 2000: Morgan Kaufmann.

[299] Kazerouni, A., Shrewsbury, B., and Padgett, C. *Using the NXT as an educational tool in computer science classes*. in *Proceedings of the 49th Annual Southeast Regional Conference*. 2011. ACM.

[300] Anderson, M. and Gavan, C. *Engaging undergraduate programming students: experiences using lego mindstorms NXT*. in *Proceedings of the 13th annual conference on Information technology education*. 2012. ACM.

[301] Pittí, K., et al. *Resources and features of robotics learning environments (RLEs) in Spain and Latin America*. in *Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality*. 2013. ACM.

[302] Jean, P., *The origin of intelligence in the child*. 1953, Routledge en Kegan Paul, London.

[303] Papert, S. and Harel, I., *Situating constructionism.* Constructionism, 1991. 36: p. 1-11.

[304] Arlegui, J., Pina, A., and Moro, M. *A PBL approach using virtual and real robots (with BYOB and LEGO NXT) to teaching learning key competences and standard curricula in primary level*. in *Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality*. 2013. ACM.

[305] Pittí, K., et al. *A new approach to robotics teaching at computer science engineering degree based on action-research*. in *Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality*. 2013. ACM.

[306] Kawell, G. and Schafer, B. *Brainstorming How to Use Lego Mindstorms EV3 in the Classroom*. in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 2015. ACM.

[307] Lui, A.K., et al., *Facilitating independent learning with Lego Mindstorms robots.* acm Inroads, 2010. 1(4): p. 49-53.

[308] Jin, K.H. and Kearns, G. *Just Enough Programming for Eight-years Olds*. in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 2015. ACM.

[309] Klassner, F. and Schafer, B. *Using the new lego MindStorms EV3 robotics platform in CS courses*. in *Proceedings of the 45th ACM technical symposium on Computer science education*. 2014. ACM.

[310] Simon, *Using a primary-school challenge in a third-year IT course*, in *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103*. 2010, Australian Computer Society, Inc.: Brisbane, Australia. p. 147-154.

[311] Benitti, F.B.V., *Exploring the educational potential of robotics in schools: A systematic review*. Computers & Education, 2012. 58(3): p. 978-988.

[312] Barker, B.S. and Ansorge, J., *Robotics as means to increase achievement scores in an informal learning environment*. Journal of Research on Technology in Education, 2007. 39(3): p. 229-243.

[313] Nugent, G., et al. *The use of digital manipulatives in k-12: Robotics, gps/gis and programming*. in *Frontiers in Education Conference, 2009. FIE'09. 39th IEEE*. 2009. IEEE.

[314] Sullivan, F.R., *Robotics and science literacy: Thinking skills, science process skills and systems understanding*. Journal of Research in Science Teaching, 2008. 45(3): p. 373-394.

[315] Hussain, S., Lindh, J., and Shukur, G., *The effect of LEGO training on pupils' school performance in mathematics, problem solving ability and attitude: Swedish data*. Educational Technology & Society, 2006. 9(3): p. 182-194.

[316] Kelleher, C. and Pausch, R., *Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers*. ACM Computing Surveys (CSUR), 2005. 37(2): p. 83-137.

[317] Powers, K., et al. *Tools for teaching introductory programming: what works?* in *ACM SIGCSE Bulletin*. 2006. ACM.

[318] Moskal, B., Lurie, D., and Cooper, S., *Evaluating the effectiveness of a new instructional approach*. ACM SIGCSE Bulletin, 2004. 36(1): p. 75-79.

[319] Sanders, D. and Dorn, B., *Classroom experience with Jeroo*. Journal of Computing Sciences in colleges, 2003. 18(4): p. 308-316.

[320] Maloney, J., et al. *Scratch: a sneak preview [education]*. in *Creating, connecting and collaborating through computing, 2004. Proceedings. Second International Conference on*. 2004. IEEE.

[321] Goldman, K.J. *A concepts-first introduction to computer science*. in *ACM SIGCSE Bulletin*. 2004. ACM.

[322] Bergin, J., *Karel universe drag & drop editor*. ACM SIGCSE Bulletin, 2006. 38(3): p. 307-307.

[323] Carlisle, M.C., et al., *RAPTOR: a visual programming environment for teaching algorithmic problem solving*. Acm Sigcse Bulletin, 2005. 37(1): p. 176-180.

[324] Chen, S. and Morris, S. *Iconic programming for flowcharts, java, turing, etc*. in *ACM SIGCSE Bulletin*. 2005. ACM.

[325] Gudmundsen, D., Olivieri, L., and Sarawagi, N., *Using visual logic©: three different approaches in different courses-general education, CS0, and CS1*. Journal of Computing Sciences in Colleges, 2011. 26(6): p. 23-29.

[326] Rich, L., Perry, H., and Guzdial, M. *A CS1 course designed to address interests of women*. in *ACM SIGCSE Bulletin*. 2004. ACM.

[327] Kay, J.S., *Teaching robotics from a computer science perspective*. Journal of Computing Sciences in Colleges, 2003. 19(2): p. 329-336.

[328] Gray, K.E. and Flatt, M. *ProfessorJ: a gradual introduction to Java through language levels*. in *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*. 2003. ACM.

[329] Portsmore, M., *Robolab: Intuitive robotic programming software to support life long learning*. APPLE Learning Technology Review, Spring/Summer, 1999.

[330] McNerney, T.S., *From turtles to Tangible Programming Bricks: explorations in physical language design*. Personal and Ubiquitous Computing, 2004. 8(5): p. 326-337.

[331] Guzdial, M., *Programming environments for novices*. Computer science education research, 2004. 2004: p. 127-154.

[332] Soloway, E., Bonar, J., and Ehrlich, K., *Cognitive strategies and looping constructs: An empirical study*. Communications of the ACM, 1983. 26(11): p. 853-860.

[333] Green, T., *Conditional program statements and their comprehensibility to professional programmers*. Journal of Occupational Psychology, 1977. 50(2): p. 93-109.

[334] Spohrer, J.C. and Soloway, E. *Putting it all together is hard for novice programmers*. in *Proceedings of the IEEE international conference on systems, man, and cybernetics*. 1985.

[335] Bruckman, A.S., *MOOSE Crossing: Construction, community, and learning in a networked virtual world for kids*. 1997, Massachusetts Institute of Technology.

[336] Pane, J.F., Myers, B.A., and Miller, L.B. *Using HCI techniques to design a more usable programming system*. in *Human Centric Computing Languages and Environments, 2002. Proceedings. IEEE 2002 Symposia on*. 2002. IEEE.

[337] Begel, A., *A Graphical Programming Language for Interacting with the World*. 1996.

[338] Kay, A., *Etoys and simstories in Squeak*. http://www/. squeakland. org/author/etoys. html, 2007.

[339] Wang, K., et al. *3D game design with programming blocks in StarLogo TNG*. in *Proceedings of the 7th international conference on Learning sciences*. 2006. International Society of the Learning Sciences.

[340] Lionet, F. and Lamoureux, Y., *Klik & Play*. Maxis, 1994.

[341] Fernaeus, Y., Kindborg, M., and Scholz, R. *Rethinking children's programming with contextual signs*. in *Proceedings of the 2006 conference on Interaction design and children*. 2006. ACM.

[342] Smith, D.C., Cypher, A., and Tesler, L., *Programming by example: novice programming comes of age*. Communications of the ACM, 2000. 43(3): p. 75-81.

[343] Stolee, K.T. and Fristoe, T. *Expressing computer science concepts through Kodu game lab*. in *Proceedings of the 42nd ACM technical symposium on Computer science education*. 2011. ACM.

[344] Harel, I., *Children designers: Interdisciplinary constructions for learning and knowing mathematics in a computer-rich school*. 1991: Ablex Publishing.

[345] Bers, M.U., et al., *Computational thinking and tinkering: Exploration of an early childhood robotics curriculum*. Computers & Education, 2014. 72: p. 145-157.

[346] Fessakis, G., Gouli, E., and Mavroudi, E., *Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study*. Computers & Education, 2013. 63: p. 87-97.

[347] Liu, C.-C., Cheng, Y.-B., and Huang, C.-W., *The effect of simulation games on the learning of computational problem solving*. Computers & Education, 2011. 57(3): p. 1907-1918.

[348] Myers, B.A. *Visual programming, programming by example, and program visualization: a taxonomy*. in *ACM SIGCHI Bulletin*. 1986. ACM.

[349] Hornecker, E. and Buur, J. *Getting a grip on tangible interaction: a framework on physical space and social interaction*. in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 2006. ACM.

[350] Ishii, H. and Ullmer, B. *Tangible bits: towards seamless interfaces between people, bits and atoms*. in *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. 1997. ACM.

[351] Marshall, P. *Do tangible interfaces enhance learning?* in *Proceedings of the 1st international conference on Tangible and embedded interaction*. 2007. ACM.

[352] Zuckerman, O., Arida, S., and Resnick, M. *Extending tangible interfaces for education: digital montessori-inspired manipulatives*. in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2005. ACM.

[353] Fernaeus, Y. and Tholander, J., *"Looking At the Computer but Doing It On Land": Children's Interactions in a Tangible Programming Space*, in *People and computers XIX—The bigger picture*. 2006, Springer. p. 3-18.

[354] Biggs, J.B., *Teaching for quality learning at university: What the student does*. 2011: McGraw-Hill Education (UK).

[355] Guzdial, M., *Does contextualized computing education help?* ACM Inroads, 2010. 1(4): p. 4-6.

[356] Cooper, S. and Cunningham, S., *Teaching computer science in context*. Acm Inroads, 2010. 1(1): p. 5-8.

[357] Dann, W.P., Cooper, S., and Pausch, R., *Learning to Program with Alice (w/CD ROM)*. 2011: Prentice Hall Press.

[358] Guzdial, M. and Ericson, B., *Introduction to computing & programming in Java: a multimedia approach*. 2007: Pearson Prentice Hall.

[359] Duchowski, A.T. and Davis, T.A. *Teaching algorithms and data structures through graphics*. in *Proc. of Eurographics*. 2007.

[360] Matzko, S. and Davis, T.A., *A graphics-based approach to data structures*. ACM SIGCSE Bulletin, 2008. 40(3): p. 109-113.

[361] Margolis, J. and Fisher, A., *Unlocking the clubhouse: Women in computing*. 2003: MIT press.

[362] Anderson, J.R., Reder, L.M., and Simon, H.A., *Situated learning and education*. Educational researcher, 1996. 25(4): p. 5-11.

[363] Guzdial, M. *A media computation course for non-majors*. in *ACM SIGCSE Bulletin*. 2003. ACM.

[364] Guzdial, M. *Exploring hypotheses about media computation*. in *Proceedings of the ninth annual international ACM conference on International computing education research*. 2013. ACM.

[365] Sloan, R.H. and Troy, P. *CS 0.5: a better approach to introductory computer science for majors*. in *ACM SIGCSE Bulletin*. 2008. ACM.

[366] Lukkarinen, A. and Sorva, J., *Classifying the tools of contextualized programming education and forms of media computation*, in *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*. 2016, ACM: Koli, Finland. p. 51-60.

[367] Bell, T.C., Witten, I.H., and Fellows, M.R., *Computer Science Unplugged: Off-line activities and games for all ages*. 1998: Citeseer.

[368] Taub, R., Ben-Ari, M., and Armoni, M., *The effect of CS unplugged on middle-school students' views of CS*. ACM SIGCSE Bulletin, 2009. 41(3): p. 99-103.

[369] Taub, R., Armoni, M., and Ben-Ari, M., *CS unplugged and middle-school students' views, attitudes, and intentions regarding CS*. ACM Transactions on Computing Education (TOCE), 2012. 12(2): p. 8.

[370] Catrambone, R., *Generalizing solution procedures learned from examples*. Journal of Experimental Psychology: Learning, Memory, and Cognition, 1996. 22(4): p. 1020.

[371] Catrambone, R., *Improving examples to improve transfer to novel problems.* Memory & Cognition, 1994. 22(5): p. 606-615.

[372] Catrambone, R., *The subgoal learning model: Creating better examples so that students can solve novel problems.* Journal of Experimental Psychology: General, 1998. 127(4): p. 355.

[373] Margulieux, L.E., *Using Subgoal Learning and Self-Explanation to Improve Programming Education.* 2016.

[374] Morrison, B.B., Margulieux, L.E., and Guzdial, M., *Subgoals, Context, and Worked Examples in Learning Computing Problem Solving*, in *Proceedings of the eleventh annual International Conference on International Computing Education Research*. 2015, ACM: Omaha, Nebraska, USA. p. 21-29.

[375] Bielaczyc, K., Pirolli, P.L., and Brown, A.L., *Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem solving.* Cognition and instruction, 1995. 13(2): p. 221-252.

[376] Renkl, A. and Atkinson, R.K., *Learning from examples: Fostering self-explanations in computer-based learning environments.* Interactive learning environments, 2002. 10(2): p. 105-119.

[377] Atkinson, R.K., et al., *Learning from examples: Instructional principles from the worked examples research.* Review of educational research, 2000. 70(2): p. 181-214.

[378] Margulieux, L.E., Guzdial, M., and Catrambone, R. *Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications*. in *Proceedings of the ninth annual international conference on International computing education research*. 2012. ACM.

[379] Kirschner, P.A., Sweller, J., and Clark, R.E., *Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching.* Educational psychologist, 2006. 41(2): p. 75-86.

[380] Margulieux, L.E. and Catrambone, R. *Improving problem solving performance in computer-based learning environments through subgoal labels*. in *Proceedings of the first ACM conference on Learning@ scale conference*. 2014. ACM.

[381] Parsons, D. and Haden, P. *Parson's programming puzzles: a fun and effective learning tool for first programming courses*. in *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*. 2006. Australian Computer Society, Inc.

[382] Denny, P., Luxton-Reilly, A., and Simon, B. *Evaluating a new exam question: Parsons problems*. in *Proceedings of the fourth international workshop on computing education research*. 2008. ACM.

[383] Ihantola, P. and Karavirta, V., *Two-dimensional parson's puzzles: The concept, tools, and first observations.* Journal of Information Technology Education, 2011. 10.

[384] Lister, R., et al. *A multi-national study of reading and tracing skills in novice programmers*. in *ACM SIGCSE Bulletin*. 2004. ACM.

[385] McCracken, M., et al., *A multi-national, multi-institutional study of assessment of programming skills of first-year CS students.* ACM SIGCSE Bulletin, 2001. 33(4): p. 125-180.

[386] Curtis, B., et al., *Software psychology: The need for an interdisciplinary program.* Proceedings of the IEEE, 1986. 74(8): p. 1092-1106.

[387] Beck, K., et al., *Manifesto for agile software development.* 2001.

[388] Andersson, R. and Bendix, L., *eXtreme teaching: A framework for continuous improvement.* Computer Science Education, 2006. 16(3): p. 175-184.

[389] Astrachan, O., Duvall, R.C., and Wallingford, E. *Bringing extreme programming to the classroom*. in *XP Universe*. 2001.

[390] Vuokko, R. and Berg, P., *Experimenting with eXtreme Teaching Method–Assessing Students' and Teachers' Experiences.* Issues in Informing Science and Information Technology, 2007. 4.

[391] Hedin, G., Bendix, L., and Magnusson, B., *Teaching software development using extreme programming*, in *Reflections on the Teaching of Programming*. 2008, Springer. p. 166-189.

[392] Hedin, G., Bendix, L., and Magnusson, B., *Teaching extreme programming to large groups of students.* Journal of Systems and Software, 2005. 74(2): p. 133-146.

[393] Du Boulay, B., *Some difficulties of learning to program.* Journal of Educational Computing Research, 1986. 2(1): p. 57-73.

[394] Moreno, A., et al. *Visualizing programs with Jeliot 3*. in *Proceedings of the working conference on Advanced visual interfaces*. 2004. ACM.

[395] Naps, T.L., et al. *Exploring the role of visualization and engagement in computer science education*. in *ACM Sigcse Bulletin*. 2002. ACM.

[396] Sorva, J. and Sirkiä, T. *UUhistle: a software tool for visual program simulation*. in *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*. 2010. ACM.

[397] Vrachnos, E. and Jimoyiannis, A. *Dave: A dynamic algorithm visualization environment for novice learners*. in *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*. 2008. IEEE.

[398] Sorva, J., *Visual program simulation in introductory programming education.* 2012: Aalto University.

[399] Griffin, J.M., *Learning by Taking Apart: Deconstructing Code by Reading, Tracing, and Debugging*, in *Proceedings of the 17th Annual Conference on Information Technology Education*. 2016, ACM: Boston, Massachusetts, USA. p. 148-153.

[400] Voorhees, R.A., *Competency‐Based learning models: A necessary future.* New directions for institutional research, 2001. 2001(110): p. 5-13.

[401] Burke, J., et al., *Criteria for describing and assessing competency programs.* Syracuse: National Consortium of Competency-Based Education and Training: Canberra, 1975.

[402] Elam, S., *Performance Based Teacher Education. What is the State of the Art?* 1971.

[403] Harris, R., Hobart, B., and Lundberg, D., *Competency-based education and training: Between a rock and a whirlpool*. 1995: Macmillan Education AU.

[404] Kelchen, R., *Who enrolls in competency‐based education? An examination of the demographics and finances of competency‐based education programs.* The Journal of Competency-Based Education, 2016. 1(1): p. 48-59.

[405] Kahlon, A., Smarzik, L., and Kohls, M. *Competency Based Education in Lower-Division Computer Science Taught at Community Colleges*. in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 2016. ACM.

[406] Soares, L., *A 'disruptive'look at competency-based education: How the innovative use of technology will transform the college experience.* Center for American Progress, 2012.

[407] Burke, J.W., *Competency based education and training*. 1989: Psychology Press.

[408] Kerka, S., *Competency-Based Education and Training. Myths and Realities.* 1998.

[409] Bray, B. and McClaskey, K., *Make learning personal: The what, who, wow, where, and why*. 2014: Corwin Press.

[410] Lu, J. *Personalized e-learning material recommender system*. in *International conference on information technology for application*. 2004. Citeseer.

[411] Zmuda, A., Ullman, D., and Curtis, G., *Learning personalized: The evolution of the contemporary classroom*. 2015: John Wiley & Sons.

[412] Hwang, G.-J., et al., *Development of a personalized educational computer game based on students' learning styles.* Educational Technology Research and Development, 2012. 60(4): p. 623-638.

[413] Graham, C., Allen, S., and Ure, D., *Blended learning: An emerging trend in education.* Encyclopedia of Distance Learning (99. 172-179). Hershey, PA: Idea Group Inc, 2005.

[414] Bourne, J., Harris, D., and Mayadas, F., *Online engineering education: Learning anywhere, anytime.* Journal of Engineering Education, 2005. 94(1): p. 131-146.

[415] Avouris, N., Tselios, N., and Tatakis, E., *Development and evaluation of a computer‐based laboratory teaching tool.* Computer Applications in Engineering Education, 2001. 9(1): p. 8-19.

[416] Stern, F., et al., *Hands-on CFD educational interface for engineering courses and laboratories.* Journal of Engineering Education, 2006. 95(1): p. 63.

[417] Khan, A.I., et al., *Study of blended learning process in education context.* International Journal of Modern Education and Computer Science, 2012. 4(9): p. 23.

[418] Choy, M., et al., *Towards blended learning of computer programming supported by an automated system.* Blended Learning, 2007. 9.

[419] Dutton, J., Dutton, M., and Perry, J., *Do online students perform as well as lecture students?* Journal of Engineering education, 2001. 90(1): p. 131.

[420] Rosenberg, J., *Assessing online student learning via Dantes test.* Virtual University Journal, 2000. 3(1): p. 1-7.

[421] Yigit, T., et al., *Evaluation of blended learning approach in computer engineering education.* Procedia-Social and Behavioral Sciences, 2014. 141: p. 807-812.

[422] Wang, F.L., et al. *Blended teaching and learning of computer programming*. in *International Conference on Web-Based Learning*. 2007. Springer.

[423] Shen, R., et al., *Mobile learning in a large blended computer science classroom: System function, pedagogies, and their impact on learning.* IEEE Transactions on Education, 2009. 52(4): p. 538-546.

[424] Network, F.L., *The four pillars of FLIP.* Retrieved May, 2014. 19: p. 2015.

[425] Schullery, N.M., Reck, R.F., and Schullery, S.E., *Toward solving the high enrollment, low engagement dilemma: A case study in introductory business.* International Journal of Business, Humanities and Technology, 2011. 1(2): p. 1-9.

[426] Toto, R. and Nguyen, H. *Flipping the work design in an industrial engineering course*. in *Frontiers in Education Conference, 2009. FIE'09. 39th IEEE*. 2009. IEEE.

[427] Ku, P.-Y., Lin, Y.-T., and Tsai, Y.-H., *Social-Media-Assisted Learning: A Case Study ofTeaching Computer Science on Facebook.* International Journal of e-Education, e-Business, e-Management and e-Learning, 2012. 2(3): p. 262.

[428] Ellison, N.B., *Social network sites: Definition, history, and scholarship.* Journal of Computer‐Mediated Communication, 2007. 13(1): p. 210-230.

[429] Foehr, U.G., Rideout, V., and Roberts, D.F., *Generation M: Media in the lives of 8-18 year-olds.* 2005.

[430] Liccardi, I., et al. *The role of social networks in students' learning experiences*. in *ACM Sigcse Bulletin*. 2007. ACM.

[431] Mori, I., *Student expectations study.* Key findings from online research and discussion evenings held in June, 2007.

[432] Attwell, G., *Personal Learning Environments-the future of eLearning?* Elearning papers, 2007. 2(1): p. 1-8.

[433] Greenhow, C. and Robelia, B., *Informal learning and identity formation in online social networks.* Learning, Media and Technology, 2009. 34(2): p. 119-140.

[434] Levin, D. and Arafeh, S., *The digital disconnect: The widening gap between Internet-savvy students and their schools.* 2002.

[435] Pempek, T.A., Yermolayeva, Y.A., and Calvert, S.L., *College students' social networking experiences on Facebook.* Journal of applied developmental psychology, 2009. 30(3): p. 227-238.

[436] Brown, J.S. and Duguid, P., *Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation.* Organization science, 1991. 2(1): p. 40-57.

[437] Cho, H., Stefanone, M., and Gay, G. *Social information sharing in a CSCL community*. in *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*. 2002. International Society of the Learning Sciences.

[438] Cho, H., et al., *Social networks, communication styles, and learning performance in a CSCL community.* Computers & Education, 2007. 49(2): p. 309-329.

[439] Haythornthwaite, C., *Building social networks via computer networks: Creating and sustaining distributed learning communities.* Building virtual communities: Learning and change in cyberspace, 2002: p. 159-190.

[440] Haythornthwaite, C. *Learning networks, crowds and communities*. in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. 2011. ACM.

[441] Lockyer, L. and Patterson, J. *Integrating social networking technologies in education: a case study of a formal learning environment*. in *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*. 2008. IEEE.

[442] Berg, J., Berquam, L., and Christoph, K., *Social Networking Technologies: A" Poke" for Campus Services.* Educause Review, 2007. 42(2).

[443] Ebersbach, A., et al., *Wiki: web collaboration*. 2008: Springer Science & Business Media.

[444] Kleiner, B., Thomas, N., and Lewis, L., *Educational Technology in Teacher Education Programs for Initial Licensure. Statistical Analysis Report. NCES 2008-040.* National Center for Education Statistics, 2007.

[445] Roblyer, M.D., et al., *Findings on Facebook in higher education: A comparison of college faculty and student uses and perceptions of social networking sites.* The Internet and higher education, 2010. 13(3): p. 134-140.

[446] Madge, C., et al., *Facebook, social integration and informal learning at university:'It is more for socialising and talking to friends about work than for actually doing work'.* Learning, Media and Technology, 2009. 34(2): p. 141-155.

[447] Karlin, S., *Examing how youths interact online.* Education Digest, 2007. 73(4): p. 6.

[448] Altbach, P.G., *Comparative higher education: Knowledge, the university, and development*. 1998: Greenwood Publishing Group.

[449] Arnove, R.F., Torres, C.A., and Franz, S., *Comparative education: The dialectic of the global and the local*. 2012: Rowman & Littlefield Publishers.

[450] Epstein, E.H., *Comparative education research: approaches and methods*. 2014, Taylor & Francis.

[451] Ben-Peretz, M., Brown, S., and Moon, B., *Routledge international companion to education*. 2004: Routledge.

[452] Johnson, L., Levine, A., and Smith, R., *The 2008 Horizon Report: Australia–New Zealand Edition. Austin, Texas: The New Media Consortium*. 2008.

[453] Richardson, W., *Blogs, wikis, podcasts, and other powerful web tools for classrooms*. 2010: Corwin Press.

[454] Wheeler, S., YEoMAnS, P., and WHEElER, D., *The good, the bad and the wiki: Evaluating student‐generated content for collaborative learning.* British journal of educational technology, 2008. 39(6): p. 987-995.

[455] Illich, I., *Deschooling society*. 1973: Harmondsworth, Middlesex.

[456] Hlynka, D., *Postmodernism in educational technology.* Update, 1996. 2002: p. 243-246.

[457] Brinton, C.G. and Chiang, M. *Social learning networks: A brief survey*. in *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*. 2014. IEEE.

[458] Paliktzoglou, V. and Suhonen, J., *Microblogging in higher education: the Edmodo case study among computer science learners in Finland.* Journal of Cases on Information Technology (JCIT), 2014. 16(2): p. 39-57.

[459] Sharples, M., *The design of personal mobile technologies for lifelong learning.* Computers & Education, 2000. 34(3): p. 177-193.

[460] Johnson, L., et al., *The NMC Horizon Report: 2015 Museum Edition*. 2015: ERIC.

[461] Johnson, L., Adams, S., and Cummins, M., *Technology Outlook for Australian Tertiary Education 2012-2017: An NMC Horizon Report Regional Analysis*. 2012: ERIC.

[462] Clark, W. and Luckin, R., *iPads in the Classroom.* What The Research Says, 2013.

[463] Willocks, B. and Redmond, P. *Evaluating a 1-to-1 iPad project: beyond rose coloured glasses*. in *Conference Proceedings of the Australian Computers in Education Conference 2014*. 2014. Australian Council for Computers in Education.

[464] Sarma, S., Brock, D.L., and Ashton, K., *The networked physical world–proposals for engineering the next generation of computing, commerce & automatic identification.* Auto-ID Center White Paper, 2000.

[465] Zhang, H. and Zhu, L. *Internet of Things: Key technology, architecture and challenging problems*. in *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*. 2011. IEEE.

[466] Kaukalias, T. and Chatzimisios, P., *Internet of Things (IoT)*, in *Encyclopedia of Information Science and Technology, Third Edition*. 2015, IGI Global. p. 7623-7632.

[467] Ericsson, L., *More than 50 billion connected devices—taking connected devices to mass market and profitability*. 2011, Tech. Rep., 2011, http://vdna/. be/publications/Wp-50-Billions. Pdf.

[468] Consortium, N.M. and Initiative, E.L., *The NMC Horizon Report: 2015 Higher Education Edition*. Austin, TX: The New Media Consortium, 2015.

[469] Kortuem, G., et al., *Educating the Internet-of-Things generation*. Computer, 2013. 46(2): p. 53-61.

[470] Mäenpää, H., et al. *Blending Problem-and Project-Based Learning in Internet of Things Education: Case Greenhouse Maintenance*. in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 2015. ACM.

[471] Nigg, C.R. and Jordan, P.J., *Commentary: It's a difference of opinion that makes a horserace…*. Health Education Research, 2005. 20(3): p. 291-293.

[472] Gagne, R.M. and Briggs, L.J., *Principles of instructional design*. 1974: Holt, Rinehart & Winston.

[473] Paas, F., Renkl, A., and Sweller, J., *Cognitive load theory and instructional design: Recent developments*. Educational psychologist, 2003. 38(1): p. 1-4.

[474] Beetham, H. and Sharpe, R., *Rethinking pedagogy for a digital age: Designing for 21st century learning*. 2013: routledge.

[475] Hannafin, M., Land, S., and Oliver, K., *Open learning environments: Foundations, methods, and models*. Instructional-design theories and models: A new paradigm of instructional theory, 1999. 2: p. 115-140.

[476] Anderson, T., *Design-based research and its application to a call centre innovation in distance education*. Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie, 2005. 31(2).

[477] McKenney, S. and Reeves, T.C., *Educational design research*, in *Handbook of research on educational communications and technology*. 2014, Springer. p. 131-140.

[478] Bannan-Ritland, B. and Baek, J., *Teacher design research: An emerging paradigm for teachers' professional development*. Handbook of design research methods in education, 2008: p. 246-262.

[479] Van den Akker, J., et al., *Educational design research*. 2006: Routledge.

[480] Wright, G.B., *Student-centered learning in higher education*. International Journal of Teaching and Learning in Higher Education, 2011. 23(1): p. 92-97.

[481] Lister, R., et al. *Differing ways that computing academics understand teaching*. in *Proceedings of the ninth Australasian conference on Computing education-Volume 66*. 2007. Australian Computer Society, Inc.

[482] Smart, K.L., Witt, C., and Scott, J.P., *Toward learner-centered teaching: An inductive approach*. Business Communication Quarterly, 2012. 75(4): p. 392-403.

[483] Lister, R., et al. *The dimensions of variation in the teaching of data structures*. in *ACM SIGCSE Bulletin*. 2004. ACM.

[484] Derry, S.J., *Cognitive schema theory in the constructivist debate*. Educational Psychologist, 1996. 31(3-4): p. 163-174.

[485] Dijkstra, S., *The integration of instructional systems design models and constructivistic design principles*. Instructional Science, 1997. 25(1): p. 1-13.

[486] Harney, O., Hogan, M.J., and Broome, B.J., *Collaborative learning: The effects of trust and open and closed dynamics on consensus and efficacy*. Social Psychology of Education, 2012: p. 1-16.

[487] Mingfei, L. and Jie, Z. *Study on the mechanisms of team learning upon knowledge transfer: a research based on social constructivism learning theory*. in *Information Management, Innovation Management and Industrial Engineering (ICIII), 2010 International Conference on*. 2010. IEEE.

[488] Ulrich Kattmann, H.G. and Michael, K., *Das Modell der Didaktischen Rekonstruktion-Ein Rahmen für naturwis-senschaftsdidaktische Forschung und Entwicklung*. Zeitschrift für Didaktik der Naturwissenschaften, 1997(3): p. 3-18.

[489] Duit, R., et al., *The Model of Educational Reconstruction–a Framework for Improving Teaching and Learning Science1*, in *Science education research and practice in Europe*. 2012, Springer. p. 13-37.

[490] Duit, R., *Science education research internationally: Conceptions, research methods, domains of research*. Eurasia Journal of Mathematics, Science & Technology Education, 2007. 3(1): p. 3-15.

[491] Heimann, P., et al., *Unterricht: Analyse und Planung*. 1972: Schroedel Hannover.

[492] Diethelm, I., Hubwieser, P., and Klaus, R. *Students, teachers and phenomena: educational reconstruction for computer science education*. in *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*. 2012. ACM.

[493] Tomlinson, C.A., *The differentiated classroom: Responding to the needs of all learners*. 2014: Ascd.

[494] Merrill, M.D., *First principles of instruction*. Educational technology research and development, 2002. 50(3): p. 43-59.

[495] Keller, J.M., *Motivational design of instruction*. Instructional design theories and models: An overview of their current status, 1983. 1(1983): p. 383-434.

[496] Becker, K., *First principles of CS instruction*. Journal of computing sciences in colleges, 2006. 22(2): p. 77-84.

[497] Gardner, J. and Belland, B.R., *A conceptual framework for organizing active learning experiences in biology instruction.* Journal of Science Education and Technology, 2012. 21(4): p. 465-475.

[498] Seehorn, D., et al., *CSTA K--12 Computer Science Standards: Revised 2011.* 2011.

[499] Denning, P.J., Martell, C.H., and Cerf, V., *Great Principles of Computing*. 2015: MIT Press.

[500] DFE, U., *National Curriculum in England: Computing Programmes of Study.* Retrieved July, 2013. 16: p. 2014.

[501] Boehm, B.W., Brown, J.R., and Lipow, M. *Quantitative evaluation of software quality*. in *Proceedings of the 2nd international conference on Software engineering*. 1976. IEEE Computer Society Press.

[502] Buck, D. and Stucki, D.J., *Design early considered harmful: graduated exposure to complexity and structure based on levels of cognitive development.* ACM SIGCSE Bulletin, 2000. 32(1): p. 75-79.

[503] Khairuddin, N.N. and Hashim, K. *Application of Bloom's taxonomy in software engineering assessments*. in *Proceedings of the 8th WSEAS International Conference on Applied Computer Science*. 2008.

[504] Alaoutinen, S. and Smolander, K. *Student self-assessment in a programming course using bloom's revised taxonomy*. in *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*. 2010. ACM.

[505] Meerbaum-Salant, O., Armoni, M., and Ben-Ari, M., *Learning computer science concepts with scratch.* Computer Science Education, 2013. 23(3): p. 239-264.

[506] Meerbaum-Salant, O. and Hazzan, O., *An Agile Constructionist Mentoring Methodology for Software Projects in the High School.* ACM Transactions on Computing Education, 2010. 9(4): p. n4.

[507] Whalley, J.L., et al. *An Australasian study of reading and comprehension skills in novice programmers, using the bloom and SOLO taxonomies*. in *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*. 2006. Australian Computer Society, Inc.

[508] Fuller, U., et al. *Developing a computer science-specific learning taxonomy*. in *ACM SIGCSE Bulletin*. 2007. ACM.

[509] Cassel, L., et al., *Computer science curriculum 2008: An interim revision of CS 2001.* 2008.

[510] Shneider, E. and Gladkikh, O., *Designing questioning strategies: Applying Bloom's taxonomy to online computer programming courses.* 2006.

[511] Bower, M. *A taxonomy of task types in computing*. in *ACM SIGCSE Bulletin*. 2008. ACM.

[512] Gluga, R., et al., *An architecture for systematic tracking of skills and competence level progression in computer science.* Computer Science Education: Innovation and Technology, 2011.

[513] Gluga, R., et al. *Coming to terms with Bloom: an online tutorial for teachers of programming fundamentals*. in *Proceedings of the Fourteenth Australasian Computing Education Conference-Volume 123*. 2012. Australian Computer Society, Inc.

[514] Krathwohl, D.R., *A revision of Bloom's taxonomy: An overview.* Theory into practice, 2002. 41(4): p. 212-218.

[515] Scott, T., *Bloom's taxonomy applied to testing in computer science classes.* Journal of Computing Sciences in Colleges, 2003. 19(1): p. 267-274.

[516] Petersen, A., Craig, M., and Zingaro, D. *Reviewing CS1 exam question content*. in *Proceedings of the 42nd ACM technical symposium on Computer science education*. 2011. ACM.

[517] Chinn, D., et al. *Introductory programming: examining the exams*. in *Proceedings of the Fourteenth Australasian Computing Education Conference-Volume 123*. 2012. Australian Computer Society, Inc.

[518] Oliver, D., et al. *This course has a Bloom Rating of 3.9*. in *Proceedings of the Sixth Australasian Conference on Computing Education-Volume 30*. 2004. Australian Computer Society, Inc.

[519] Biggs, J.B. and Collis, K.F., *Evaluation the quality of learning: the SOLO taxonomy (structure of the observed learning outcome).* 1982: Academic Press.

[520] Biggs, J. and Tang, C., *Teaching for quality learning at university (Society for research into higher education).* 2007.

[521] Atherton, J., *Learning and Teaching; SOLO taxonomy (2005).*

[522] Jimoyiannis, A., *Using SOLO taxonomy to explore students' mental models of the programming variable and the assignment statement.* Themes in Science and Technology Education, 2013. 4(2): p. 53-74.

[523] Lister, R., et al., *Not seeing the forest for the trees: novice programmers and the SOLO taxonomy.* ACM SIGCSE Bulletin, 2006. 38(3): p. 118-122.

[524] Sheard, J., et al. *Going SOLO to assess novice programmers*. in *ACM SIGCSE Bulletin*. 2008. ACM.

[525] Lister, R., Fidge, C., and Teague, D. *Further evidence of a relationship between explaining, tracing and writing skills in introductory programming*. in *ACM SIGCSE Bulletin*. 2009. ACM.

[526] Whalley, J., et al. *Salient elements in novice solutions to code writing problems*. in *Proceedings of the Thirteenth Australasian Computing Education Conference-Volume 114*. 2011. Australian Computer Society, Inc.

[527] Clear, T., et al., *Report on the final BRACElet workshop.* Journal of Applied Computing and Information Technology, 2011. 15(1): p. 2011.

[528] Brabrand, C. and Dahl, B., *Using the SOLO taxonomy to analyze competence progression of university science curricula.* Higher Education, 2009. 58(4): p. 531-549.

[529] Roblyer, M.D., Edwards, J., and Havriluk, M.A., *Integrating educational technology into teaching*. Vol. 2. 2006: Pearson/Merrill Prentice Hall Upper Saddle River, NJ.

[530] Koehler, M.J. and Mishra, P., *What is technological pedagogical content knowledge.* Contemporary issues in technology and teacher education, 2009. 9(1): p. 60-70.

[531] Hughes, J., Thomas, R., and Scharber, C., *Assessing technology integration: The RAT-replacement, amplification, and transformation-framework.* Technology and Teacher Education Annual, 2006. 3: p. 1616.

[532] Allsopp, M.M., Hohlfeld, T., and Kemker, K., *The Technology Integration Matrix: The development and field-test of an Internet based multi-media assessment tool for the implementation of instructional technology in the classroom.* Florida Educational Research Association, Tampa, 2007.

[533] Puentedura, R., *The SAMR model: Background and exemplars.* Retrieved June, 2012. 24: p. 2013.

[534] Farance, F. and Tonkel, J., *Learning Technology Systems Architecture (LTSA).* IEEE Learning Technology Standards Committee (LTSC), IEEE P, 1999. 1484.

[535] Wiley, D.A., *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy.* 2003.

[536] Sokolova, K., *Alternative teaching methods in Informatics and general public attitudes about them*, in *Departement of Informatics and Telecommunications*. 2016, University of Peloponnese: Tripolis. p. 149.

[537] Drachsler, H. and Kirschner, P.A., *Learner Characteristics*, in *Encyclopedia of the Sciences of Learning*, N.M. Seel, Editor. 2012, Springer US: Boston, MA. p. 1743-1745.

[538] Arends, R., *Learning to teach*. 2014: McGraw-Hill Higher Education.

[539] Clotfelter, C.T., Ladd, H.F., and Vigdor, J.L., *Teacher credentials and student achievement in high school: A cross-subject analysis with student fixed effects*. 2007, National Bureau of Economic Research.

[540] Walvoord, B.E., *Assessment clear and simple: A practical guide for institutions, departments, and general education*. 2010: John Wiley & Sons.

[541] Ormrod, J., *How motivation affects learning and behavior*. 2010, Pearson Allyn Bacon Prentice Hall, Boston.

[542] Fredricks, J.A., Blumenfeld, P.C., and Paris, A.H., *School engagement: Potential of the concept, state of the evidence.* Review of educational research, 2004. 74(1): p. 59-109.

[543] Metcalfe, J.E. and Shimamura, A.P., *Metacognition: Knowing about knowing*. 1994: The MIT Press.

[544] Lias, T.E. and Elias, T., *Learning analytics: The definitions, the processes, and the potential.* 2011.

[545] Siemens, G. and Long, P., *Penetrating the fog: Analytics in learning and education.* EDUCAUSE review, 2011. 46(5): p. 30.

[546] Bonwell, C.C. and Eison, J.A., *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports*. 1991: ERIC.

[547] Felder, R.M., et al., *The future of engineering education II. Teaching methods that work.* Chemical Engineering Education, 2000. 34(1): p. 26-39.

[548] Grissom, S., *Introduction to special issue on alternatives to lecture in the computer science classroom.* Trans. Comput. Educ., 2013. 13(3): p. 1-5.

[549] Council, N.R., *Education for life and work: Developing transferable knowledge and skills in the 21st century*. 2013: National Academies Press.

[550] Lombardi, M.M., *Authentic learning for the 21st century: An overview.* Educause learning initiative, 2007. 1(2007): p. 1-12.

[551] Saavedra, A.R. and Opfer, V.D., *Teaching and Learning 21st Century Skills.* 2012.

[552] Michael, J. and Modell, H.I., *Active learning in secondary and college science classrooms: A working model for helping the learner to learn*. 2003: Routledge.

[553] Dagdilelis, V., Satratzemi, M., and Evangelidis, G., *Introducing secondary education students to algorithms and programming.* Education and Information Technologies, 2004. 9(2): p. 159-173.

[554] Baran, B., *Facebook as a formal instructional environment.* British Journal of Educational Technology, 2010. 41(6): p. E146-E149.

[555] Ebert-May, D., Brewer, C., and Allred, S., *Innovation in large lectures: Teaching for active learning.* Bioscience, 1997. 47(9): p. 601-607.

[556] Laws, P., Sokoloff, D., and Thornton, R., *Promoting active learning using the results of physics education research.* UniServe Science News, 1999. 13: p. 14-19.

[557] Sarason, Y. and Banbury, C., *Active learning facilitated by using a game-show format or who doesn't want to be a millionaire?* Journal of Management Education, 2004. 28(4): p. 509-518.

[558] Smith, B.L. and MacGregor, J.T., *What is collaborative learning.* 1992.

[559] Gerlach, J.M., *Is this collaboration?* New directions for teaching and learning, 1994. 1994(59): p. 5-14.

[560] Dillenbourg, P., *What do you mean by collaborative learning.* Collaborative-learning: Cognitive and computational approaches, 1999. 1: p. 1-15.

[561] Roschelle, J. and Teasley, S.D. *The construction of shared knowledge in collaborative problem solving*. in *Computer supported collaborative learning*. 1995. Springer.

[562] Veerman, A., Andriessen, J., and Kanselaar, G., *Collaborative argumentation in academic education.* Instructional Science, 2002. 30(3): p. 155-186.

[563] Kirschner, F., Paas, F., and Kirschner, P.A., *A cognitive load approach to collaborative learning: United brains for complex tasks.* Educational Psychology Review, 2009. 21(1): p. 31-42.

[564] Terenzini, P.T., et al., *Collaborative learning vs. lecture/discussion: Students' reported learning gains.* Journal of Engineering Education, 2001. 90(1): p. 123.

[565] Bergmann, J. and Sams, A., *Flip your classroom: Reach every student in every class every day*. 2012: International Society for Technology in Education.

[566] Whitman, N.A. and Fife, J.D., *Peer Teaching: To Teach Is To Learn Twice. ASHE-ERIC Higher Education Report No. 4, 1988*. 1988: ERIC.

[567] Ploetzner, R., et al., *Learning by explaining to oneself and to others.* Collaborative learning: Cognitive and computational approaches, 1999: p. 103-121.

[568] Ballantyne, R., Hughes, K., and Mylonas, A., *Developing procedures for implementing peer assessment in large classes using an action research process.* Assessment & Evaluation in Higher Education, 2002. 27(5): p. 427-441.

[569] Dochy, F., Segers, M., and Sluijsmans, D., *The use of self-, peer and co-assessment in higher education: A review.* Studies in Higher Education, 1999. 24(3): p. 331-350.

[570] Topping, K.J., et al., *Formative peer assessment of academic writing between postgraduate students.* Assessment & Evaluation in Higher Education, 2000. 25(2): p. 149-169.

[571] Van den Berg, I., Admiraal, W., and Pilot, A., *Design principles and outcomes of peer assessment in higher education.* Studies in Higher Education, 2006. 31(03): p. 341-356.

[572] Chang, T.-Y. and Chen, Y.-T., *Cooperative learning in E-learning: A peer assessment of student-centered using consistent fuzzy preference.* Expert Systems with Applications, 2009. 36(4): p. 8342-8349.

[573] Gouli, E., Gogoulou, A., and Grigoriadou, M., *Supporting self-, peer-, and collaborative-assessment in e-learning: the case of the PEer and Collaborative ASSessment Environment (PECASSE).* Journal of Interactive Learning Research, 2008. 19(4): p. 615.

[574] Aiken, R.M., Bessagnet, M.-N., and Israel, J., *Interaction and Collaboration Using an Intelligent Collaborative Learning Environment.* Education and Information Technologies, 2005. 10(1-2): p. 67-82.

[575] Beck, L. and Chizhik, A., *Cooperative learning instructional methods for CS1: Design, implementation, and evaluation.* Trans. Comput. Educ., 2013. 13(3): p. 1-21.

[576] McManus, M.M. and Aiken, R.M., *Teaching collaborative skills with a group leader computer tutor.* Education and Information Technologies, 1996. 1(1): p. 75-96.

[577] Bouras, C. and Paraskevas, M. *Educational Information Society in Greece: The Greek School Network*. in *IADIS International Conference e–SOCIETY, Lisbon, Portugal*. 2003. Citeseer.

[578] Parham, J., Chinn, D., and Stevenson, D. *Using Bloom's taxonomy to code verbal protocols of students solving a data structure problem*. in *Proceedings of the 47th Annual Southeast Regional Conference*. 2009. ACM.

[579] Cartwright, D.E. and Zander, A.E., *Group dynamics research and theory.* 1953.

[580] Engleberg, I.N. and Wynn, D., *Working in groups*. 2000: Houghton Mifflin Company.

[581] Berry Jr, L., *Collaborative Learning: A Program for Improving the Retention of Minority Students.* 1991.

[582] Johnson, D.W. and Johnson, R.T., *Mainstreaming and cooperative learning strategies.* Exceptional children, 1986. 52(6): p. 530-561.

[583] Rea, L.M. and Parker, R.A., *Designing and conducting survey research: A comprehensive guide*. 2014: John Wiley & Sons.

[584] McCarney, R., et al., *The Hawthorne Effect: a randomised, controlled trial.* BMC medical research methodology, 2007. 7(1): p. 30.

[585] Wright, K.D. and Asmundson, G.J., *Health anxiety in children: Development and psychometric properties of the Childhood Illness Attitude Scales.* Cognitive Behaviour Therapy, 2003. 32(4): p. 194-202.

[586] Mellor, D. and Moore, K.A., *The use of Likert scales with children.* Journal of pediatric psychology, 2014. 39(3): p. 369-379.

[587] Felder, R.M. and Brent, R., *Navigating the bumpy road to student-centered instruction.* College teaching, 1996. 44(2): p. 43-47.

[588] Felder, R.M. and Brent, R., *Learning by doing.* Chemical engineering education, 2003. 37(4): p. 282-309.

[589] Michel, N., Cater, J.J., and Varela, O., *Active versus passive teaching styles: An empirical study of student learning outcomes.* Human Resource Development Quarterly, 2009. 20(4): p. 397-418.

[590] Sprenger, M., *Differentiation through learning styles and memory*. 2008: Corwin Press.

[591] Hawk, T.F. and Shah, A.J., *Using learning style instruments to enhance student learning.* Decision Sciences Journal of Innovative Education, 2007. 5(1): p. 1-19.

[592] Williams, A.L. and Merten, M.J., *A review of online social networking profiles by adolescents: Implications for future research and intervention.* Adolescence, 2008. 43(170): p. 253.

[593] Oradini, F. and Saunders, G. *The use of social networking by students and staff in higher education*. in *iLearning Forum*. 2008.

[594] Boyd, D. and Ellison, N., *Social network sites: definition, history, and scholarship.* IEEE Engineering Management Review, 2010. 3(38): p. 16-31.

[595] Dalsgaard, C., *Social software: E-learning beyond learning management systems.* European Journal of Open, Distance and E-Learning, 2006. 9(2).

[596] Klamma, R., et al., *Social software for life-long learning.* Educational technology & society, 2007. 10(3): p. 72-83.

[597] Crook, C. and Cluley, R., *The teaching voice on the learning platform: seeking classroom climates within a virtual learning environment.* Learning, Media and Technology, 2009. 34(3): p. 199-213.

[598] Spires, H.A., et al., *Having our say: Middle grade student perspectives on school, technologies, and academic engagement.* Journal of Research on Technology in Education, 2008. 40(4): p. 497-515.

[599] Barron, B., *Interest and self-sustained learning as catalysts of development: A learning ecology perspective.* Human development, 2006. 49(4): p. 193-224.

[600] Selwyn, N., *Faceworking: exploring students' education‐related use of Facebook.* Learning, Media and Technology, 2009. 34(2): p. 157-174.

[601] Roblyer, M. and Wiencke, W., *Exploring the interaction equation: Validating a rubric to assess and encourage interaction in distance courses.* Journal of Asynchronous Learning Networks, 2004. 8(4): p. 24-37.

[602] Mazer, J.P., Murphy, R.E., and Simonds, C.J., *I'll see you on "Facebook": The effects of computer-mediated teacher self-disclosure on student motivation, affective learning, and classroom climate.* Communication Education, 2007. 56(1): p. 1-17.

[603] Ellison, N.B., Steinfield, C., and Lampe, C., *The benefits of Facebook "friends:" Social capital and college students' use of online social network sites.* Journal of Computer‐Mediated Communication, 2007. 12(4): p. 1143-1168.

[604] Witkin, H., *Psychological differention; Studies of development*. 1962, Wiley, New York.

[605] Witkin, H.A., *Cognitive styles in personal and cultural adaptation*. 1978: Clark Univ Heinz Werner Inst.

[606] Kagan, J., *Impulsive and reflective children: Significance of conceptual tempo.* Learning and the educational process. Chicago: Rand McNally, 1965. 133: p. 16l.

[607] Hudson, L., *Contrary imaginations, by liam hudson*. 1966: Methuen.

[608] Pask, G., *A fresh look at cognition and the individual.* International Journal of Man-Machine Studies, 1972. 4(3): p. 211-216.

[609] Alavi, M., *Computer-mediated collaborative learning: An empirical evaluation.* MIS quarterly, 1994: p. 159-174.

[610] Johnson, S.D., et al., *Comparative analysis of learner satisfaction and learning outcomes in online and face-to-face learning environments.* Journal of interactive learning research, 2000. 11(1): p. 29.

[611] Loomis, K.D., *Learning styles and asynchronous learning: Comparing the LASSI model to class performance.* Journal of Asynchronous Learning Networks, 2000. 4(1): p. 23-32.

[612] Parker, D. and Gemino, A., *Inside online learning: Comparing conceptual and technique learning performance in place-based and ALN formats.* Journal of Asynchronous Learning Networks, 2001. 5(2): p. 64-74.

[613] Acquisti, A. and Gross, R. *Imagined communities: Awareness, information sharing, and privacy on the Facebook*. in *International workshop on privacy enhancing technologies*. 2006. Springer.

[614] Barnes, S.B., *A privacy paradox: Social networking in the United States.* First Monday, 2006. 11(9).

[615] Peluchette, J. and Karl, K., *Social networking profiles: An examination of student attitudes regarding use and appropriateness of content.* CyberPsychology & Behavior, 2008. 11(1): p. 95-97.

[616] Haythornthwaite, C., *Social networks and Internet connectivity effects.* Information, Community & Society, 2005. 8(2): p. 125-147.

[617] Mack, D., et al., *Reaching students with Facebook: data and best practices.* Electronic journal of academic and special librarianship, 2007. 8(2): p. 4.

[618] Haythornthwaite, C., et al., *Community development among distance learners: Temporal and technological dimensions.* Journal of Computer‐Mediated Communication, 2000. 6(1): p. 0-0.

[619] Suzuki, L.K. and Calzo, J.P., *The search for peer advice in cyberspace: An examination of online teen bulletin boards about health and sexuality.* Journal of applied developmental psychology, 2004. 25(6): p. 685-698.

[620] Martin, B. and Briggs, L., *The cognitive and affective domains: Integration for instruction and research.* Englewood Cliffs, NJ: Educational Technology Publications, 1986. 35: p. 123-130.

[621] Miller, M., *Teaching and learning in affective domain.* Emerging perspectives on learning, teaching, and technology. Retrieved March, 2005. 6: p. 2008.

[622] Underwood, J., et al., *Impact 2008.* Coventry: Becta, 2009.

[623] Hughes, G., *Social software: new opportunities for challenging social inequalities in learning?* Learning, media and technology, 2009. 34(4): p. 291-305.

[624] Djaouti, D., Alvarez, J., and Jessel, J.-P., *Classifying serious games: The G/P/S model.* Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches, 2011: p. 118-136.

[625] Hanks, B. *Student attitudes toward pair programming*. in *ACM SIGCSE Bulletin*. 2006. ACM.

[626] Howard, E.V., *Attitudes on using pair-programming.* Journal of Educational Technology Systems, 2006. 35(1): p. 89-103.

[627] Begel, A. and Nagappan, N. *Pair programming: what's in it for me?* in *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*. 2008. ACM.

[628] Williams, L.A. and Kessler, R.R., *Experiments with industry's "pair-programming" model in the computer science classroom.* Computer Science Education, 2001. 11(1): p. 7-20.

[629] Cooper, D.J. and Kagel, J.H., *Are two heads better than one? Team versus individual play in signaling games.* American Economic Review, 2005: p. 477-509.

[630] Ames, C. and Archer, J., *Achievement goals in the classroom: Students' learning strategies and motivation processes.* Journal of educational psychology, 1988. 80(3): p. 260.

[631] Theodoropoulos, A., Antoniou, A., and Lepouras, G., *Students teach students: Alternative teaching in Greek secondary education.* Education and Information Technologies, 2014: p. 1-27.

[632] Fagin, B.S., Merkle, L.D., and Eggers, T.W., *Teaching computer science with robotics using Ada/Mindstorms 2.0.* ACM SIGAda Ada Letters, 2001. 21(4): p. 73-78.

[633] Armoni, M., Meerbaum-Salant, O., and Ben-Ari, M., *From scratch to "real" programming.* ACM Transactions on Computing Education (TOCE), 2015. 14(4): p. 25.

[634] McGill, M.M., *Learning to program with personal robots: Influences on student motivation.* ACM Transactions on Computing Education (TOCE), 2012. 12(1): p. 4.

[635] Lu, D.V. and Mead, R. *Introducing students grades 6–12 to expressive robotics*. in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*. 2012. IEEE.

[636] Rusk, N., et al., *New pathways into robotics: Strategies for broadening participation.* Journal of Science Education and Technology, 2008. 17(1): p. 59-69.

[637] Apiola, M., Lattu, M., and Pasanen, T.A. *Creativity and intrinsic motivation in computer science education: experimenting with robots*. in *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*. 2010. ACM.

[638] Melchior, A., et al., *More than robots: An evaluation of the first robotics competition participant and institutional impacts.* Heller School for Social Policy and Management, Brandeis University, 2005.

[639] Melchior, A., Cutter, T., and Cohen, F., *Evaluation of FIRST LEGO league.* Waltham, MA: Center for Youth and Communities, Heller Graduate School, Brandeis University, 2004.

[640] Giannakos, M.N. and Jaccheri, L. *What motivates children to become creators of digital enriched artifacts?* in *Proceedings of the 9th ACM Conference on Creativity & Cognition*. 2013. ACM.

[641] Hubwieser, P., *The darmstadt model: a first step towards a research framework for computer science education in schools*, in *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages*. 2013, Springer. p. 1-14.

[642] Hubwieser, P., et al., *Computer science/informatics in secondary education*, in *Proceedings of the 16th annual conference reports on Innovation and technology in computer science education - working group reports*. 2011, ACM: Darmstadt, Germany. p. 19-38.

[643] Pintrich, P.R. and De Groot, E.V., *Motivational and self-regulated learning components of classroom academic performance.* Journal of educational psychology, 1990. 82(1): p. 33.

[644] Glynn, S.M., et al., *Science motivation questionnaire II: Validation with science majors and nonscience majors.* Journal of research in science teaching, 2011. 48(10): p. 1159-1176.

[645] Nugent, G., et al., *Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes.* Journal of Research on Technology in Education, 2010. 42(4): p. 391-408.

[646] Cruz, I.L.J., *Robotics as a means of increasing student achievement in middle school science*. 2010, Faculty of the Louisiana State University and Agricultural and Mechanical College in partial fulfillment of the requirements for the degree of Master of Natural Sciences in The Interdepartmental Program in Natural Sciences By Ingrid Lorelei J. Cruz BS, University of the Philippines.

[647] Mayring, P. *On generalization in qualitatively oriented research*. in *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*. 2007.

[648] Catlin, D. and Woollard, J. *Educational robots and computational thinking*. in *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education*. 2014.

[649] Bourgonjon, J., et al., *Students' perceptions about the use of video games in the classroom.* Computers & Education, 2010. 54(4): p. 1145-1156.

[650] Giannakos, M.N., *Enjoy and learn with educational games: Examining factors affecting learning performance.* Comput. Educ., 2013. 68: p. 429-439.

[651] Bishop-Clark, C., *Cognitive style, personality, and computer programming.* Computers in Human Behavior, 1995. 11(2): p. 241-260.

[652] Da Cunha, A.D. and Greathead, D., *Does personality matter?: an analysis of code-review ability.* Communications of the ACM, 2007. 50(5): p. 109-112.

[653] Werner, L., et al. *Can middle-schoolers use Storytelling Alice to make games?: results of a pilot study*. in *Proceedings of the 4th International Conference on Foundations of Digital Games*. 2009. ACM.

[654] Werner, L., Denner, J., and Campe, S., *Children programming games: a strategy for measuring computational learning.* ACM Transactions on Computing Education (TOCE), 2014. 14(4): p. 24.

[655] Repenning, A., et al., *Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation.* Trans. Comput. Educ., 2015. 15(2): p. 1-31.

[656] Jiau, H.C., Chen, J.C., and Ssu, K.-F., *Enhancing self-motivation in learning programming using game-based simulation and metrics.* Education, IEEE Transactions on, 2009. 52(4): p. 555-562.

[657] Zualkernan, I., Allert, J., and Qadah, G.Z., *Learning styles of computer programming students: a Middle Eastern and American comparison.* Education, IEEE Transactions on, 2006. 49(4): p. 443-450.

[658] Hubwieser, P., Armoni, M., and Giannakos, M.N., *How to Implement Rigorous Computer Science Education in K-12 Schools? Some Answers and Many Questions.* Trans. Comput. Educ., 2015. 15(2): p. 1-12.

[659] Malinovski, T., Vasileva, M., and Trajkovik, V., *Integrating Computer Games in Primary Education for Increased Students' QoE*, in *ICT Innovations 2013*. 2014, Springer. p. 35-44.

[660] Laghari, K.U.R. and Connelly, K., *Toward total quality of experience: A QoE model in a communication ecosystem.* Communications Magazine, IEEE, 2012. 50(4): p. 58-65.

[661] Giannakos, M., Hubwieser, P., and Chrisochoides, N., *How students estimate the effects of ICT and programming courses*, in *Proceeding of the 44th ACM technical symposium on Computer science education*. 2013, ACM: Denver, Colorado, USA. p. 717-722.

[662] Makris, D., et al., *Could you help me to change the variables?: comparing instruction to encouragement for teaching programming*, in *Proceedings of the 8th Workshop in Primary and Secondary Computing Education*. 2013, ACM: Aarhus, Denmark. p. 79-82.

[663] Dickson, P.E., *Using Unity to Teach Game Development: When You've Never Written a Game*, in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. 2015, ACM: Vilnius, Lithuania. p. 75-80.

[664] Davis, F.D., *Perceived usefulness, perceived ease of use, and user acceptance of information technology.* MIS Q., 1989. 13(3): p. 319-340.

[665] Chamillard, A.T., *Using a student response system in CS1 and CS2*, in *Proceedings of the 42nd ACM technical symposium on Computer science education*. 2011, ACM: Dallas, TX, USA. p. 299-304.

[666] Venkatesh, V. and Bala, H., *Technology acceptance model 3 and a research agenda on interventions.* Decision sciences, 2008. 39(2): p. 273-315.

[667] Boyce, A.K., et al. *Experimental evaluation of BeadLoom game: how adding game elements to an educational tool improves motivation and learning*. in *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*. 2011. ACM.

[668] Preist, C. and Jones, R., *The Use of Games as Extrinisic Motivation in Education*, in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, ACM: Seoul, Republic of Korea. p. 3735-3738.

[669] Leblanc, S., et al., *An analysis of a user's exploration and learning of a multimedia instruction system.* Computers & Education, 2001. 36(1): p. 59-82.

[670] Witkin, H.A., *Psychological differention; Studies of development*. 1962, Wiley, New York.

[671] Ford, N., *Levels and types of mediation in instructional systems: an individual differences approach.* International Journal of Human-Computer Studies, 1995. 43(2): p. 241-259.

[672] Gardner, H., *Frames of mind: The theory of multiple intelligences*. 2011: Basic books.

[673] Kyllonen, P.C. and Christal, R.E., *Reasoning ability is (little more than) working-memory capacity?!* Intelligence, 1990. 14(4): p. 389-433.

[674] Biggs, J.B., *Student Approaches to Learning and Studying. Research Monograph*. 1987: ERIC.

[675] Elton, L., *Strategies to enhance student motivation: a conceptual analysis.* Studies in Higher Education, 1996. 21(1): p. 57-68.

[676] Atkins, H., et al., *Learning Style Theory and Computer Mediated Communication.* 2001.

[677] Myers, I.B., McCaulley, M.H., and Most, R., *Manual, a guide to the development and use of the Myers-Briggs type indicator*. 1985: Consulting Psychologists Press.

[678] Batista, I.V. and Cornachione, E.B., *Learning styles influences on satisfaction and perceived learning: Analysis of an online business game.* Developments in Business Simulation and Experiential Learning, 2014. 32.

[679] Sung, H.-Y., et al. *Effect of Learning Styles on Students' Motivation and Learning Achievement in Digital Game-based Learning*. in *Advanced Applied Informatics (IIAIAAI), 2012 IIAI International Conference on*. 2012. IEEE.

[680] Lau, W.W. and Yuen, A.H., *Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy.* British Journal of Educational Technology, 2009. 40(4): p. 696-712.

[681] Milovanović, M., et al., *Effectiveness of game-based learning: Influence of cognitive style*, in *Best Practices for the Knowledge Society. Knowledge, Learning, Development and Technology for All*. 2009, Springer. p. 87-96.

[682] Shu, N.C., *Visual programming*. 1988: Van Nostrand Reinhold New York.

[683] Myers, B.A., *Taxonomies of visual programming and program visualization.* Journal of Visual Languages & Computing, 1990. 1(1): p. 97-123.

[684] Resnick, M., et al., *Scratch: programming for all.* Communications of the ACM, 2009. 52(11): p. 60-67.

[685] Child, D., *The relationships between introversion‐extraversion, neuroticism and performance in school examinations.* British Journal of Educational Psychology, 1964. 34(2): p. 187-196.

[686] Martin, C.R., *Looking at type: The fundamentals*. 1997: Center for Applications of Psychological Type.

[687] Lawrence, G.D., *Looking at type and learning styles*. 1997: Center for Applications of Psychological Type, Incorporated.

[688] Farrell, D. and Moffat, D.C., *Adapting Cognitive Walkthrough to Support Game Based Learning Design.* International Journal of Game-Based Learning (IJGBL), 2014. 4(3): p. 23-34.

[689] Aleven, V., et al. *Toward a framework for the analysis and design of educational games*. in *Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2010 Third IEEE International Conference on*. 2010. IEEE.

[690] Squire, K., et al., *Design Principles of Next-Generation Digital Gaming for Education*. Educational Technology, 2003. 43(5): p. 17-23.

[691] Ross, S.M. and Morrison, G.R., *Experimental research methods*. Handbook of research for educational communications and technology: A project of the association for educational communications and technology, 1996: p. 1148-1170.

[692] Williams, L., et al., *Strengthening the case for pair programming*. IEEE software, 2000. 17(4): p. 19-25.

[693] Lai, H. and Xin, W. *An experimental research of the pair programming in java programming course*. in *e-Education, Entertainment and e-Management (ICEEE), 2011 International Conference on*. 2011. IEEE.

[694] Ramli, N. and Fauzi, S.S.M. *The effects of pair programming in programming language subject*. in *2008 International Symposium on Information Technology*. 2008. IEEE.