Increasing computer science popularity and gender diversity through the use of games and contextualized learning

Submitted in partial fulfilment
of the requirements of the degree of

Bachelor of Science (Honours)

of Rhodes University

Mikha Zeffertt
Supervisors: M. Halse and H. Thinyane

Grahamstown, South Africa
31 October 2014

Abstract

There has been an international decline in university enrolment rates for computer science courses in recent years. There also exists a distinct gender divide in the subject, with female students exhibiting disinterest in a useful and interesting field due to a wide variety of reasons. In order to try to counteract these two deficiencies in computer science, the way the subject is being taught and presented to students should be revisited and improved upon. The use of educational games to teach basic computer science skills in contextualized and interactive ways could generate enthusiasm and interest for the subject. Both contextualized learning and the use of games for teaching have been shown to be gender neutral. Research has revealed that females have low self efficacy levels in regard to computers. By presenting students with an easy to play computational thinking game, the confidence levels amongst students, females especially, could be increased. An adventure style game was developed to be used by Computer Science 112 (CSc 112) students at Rhodes University during their computational thinking module. By researching teaching and game trends, the game was specifically designed to appeal to both genders. Pre- and post-intervention surveys were run to assess the change in CSc 112 students' attitudes towards themselves and the subject.

The surveys indicated that the mean female responses rating their own abilities, confidence and interest in computer science decreased significantly. Mean male responses were less consistent, with the mean rating of male confidence increasing but their desire to succeed and interest in computer science decreasing. The exact reason for this change in attitudes could not be isolated. The developed game was well received by students, with the majority of questioned students liking contextualized learning as well as learning through the use of games. A higher percentage of students claimed the game increased their problem solving confidence than the percentage that said the same about regular CSc 112 practicals. The game proved to be gender neutral amongst the CSc 112 students.

ACM Computing Classification System Classification

ACM Computing Classification System Classification (2012 version, valid through 2014)

L.10.3 [Applied computing] Interactive learning environments
M.1.4.1 [Social and professional topics] Computational thinking
M.1.4.4 [Social and professional topics] Computing education programs
M.3.3 [Social and professional topics] Gender

General Terms: Computational thinking, gender, educational games, contextualized learning

Acknowledgments

I would like to thank the National Research Foundation and Rhodes University for the financial support that allowed me to carry out this research.

This work was undertaken in the Distributed Multimedia CoE at Rhodes University, with financial support from Telkom SA, Tellabs, Genband, Easttel, Bright Ideas 39, THRIP and NRF SA (TP13070820716). The authors acknowledge that opinions, findings and conclusions or recommendations expressed here are those of the author(s) and that none of the above mentioned sponsors accept liability whatsoever in this regard.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem definition

Formal computer science education is becoming less and less popular around the world, particularly amongst female students (Bayliss, 2009, Mitchell, 2013). The exact shape of the gender decline can be seen in Figure 1.1, where in 2010 females were making up less than 20% of computer science bachelor's graduates in the United States of America. Although this case is not identical in situation to the rest of the world, it can be seen as a reflection of the gender imbalance in the field. In order to try and counteract this, the way computer science is being taught and presented to students needs to be revisited and improved upon. The use of educational games to teach basic computer science skills in contextualized and interactive ways could generate enthusiasm and interest for the subject. This approach can be tailored towards a gender neutral teaching method, in order to be make the subject more inclusive and approachable for all students. Teaching computational thinking through situational problem solving will place what the students learn in context, as well as help reveal the relevance of these skills. The abstract nature of computer science causes many females to shy away from the field, but a contextualized approach has been shown to increase students' interest and investment in subject matter, regardless of gender (Carter, 2006). Prior research has revealed that females feel less comfortable than males when interacting with computers, for various reasons (Beyer, Rynes, Perrault, Hay, & Haller, 2003). By presenting these students with an easy to play game, the confidence levels amongst students, particularly females, could be increased, for both computer interaction and problem solving (Carbonaro, Szafron, Cutumisu, & Schaeffer, 2010).

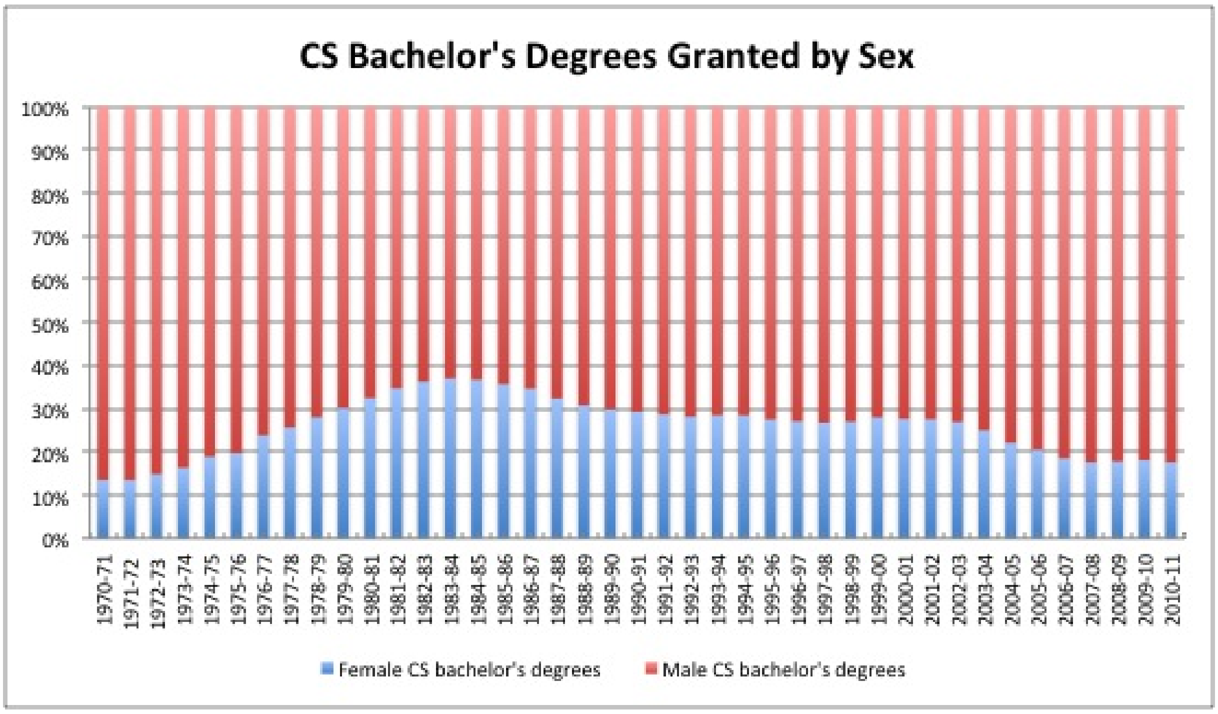

Figure 1.1: Graph of percentage of computer science bachelor's degrees from 1970 to 2010 in the United States of America (Mitchell, 2013)

An adventure style game was developed to teach basic computational thinking skills with increasing levels of difficulty to be used during a computer science introductory course, within the programming logic module. By researching teaching techniques and game trends, the game is specifically designed to appeal to both genders. Through continuous encouragement, veiled assistance and enjoyment, computer science can hopefully be re-evaluated by students as a fascinating field that is worth becoming involved in. The difference between computer science and computational thinking has been acknowledged by several authors (Lu & Fletcher, 2009, Wing, 2008), yet almost all computer-science-related educational games have been designed to teach programming basics (Bayliss, 2009, Cordova & Lepper, 1996, Leutenegger & Edgington, 2007). There exists a space for a game which develops basic computational thinking skills in a gender conscious way. These skills would be useful to more than just computer science students. They would benefit any student doing a subject that involves logical problem solving or information processing (Lu & Fletcher, 2009, Wing, 2008).

This thesis explores the effect that using a game to learn basic computational thinking skills has on the students' self efficacy and attitude towards the subject, specifically its effect on females. It also explores the difference in opinions and attitudes between genders in first year computer science students, in order to identify how female students can be further encouraged and what effect the course is currently having on students.

1.2 Methodology

In order to assess the opinions and beliefs of contextualized learning and the gender divide within the field, two initial surveys were conducted at Rhodes University. The first was run within the department and was open to all students from second year up to lecturers. The second was run at the beginning of the Computer Science 112 (here after referred to as CSc 112) course. CSc 112 is a service course taken by students from mostly commerce backgrounds, but also students from the Science or Arts faculties. This particular course was selected due to time constraints (the course runs in the second semester so allowed time for game development) and because it is an introductory computer science course where students have little to no previous computer science experience.

Both initial surveys were informed by the literature review (see Chapter 2) and in turn informed the game development. The second survey was based on a previously conducted computer science attitude survey (Wiebe & Miller, 2003). The same questions from the attitude survey were asked again in a final survey that was run a week after the use of the game to assess if there had been a shift in perception of self and of the field. As the two surveys were run three months apart and encompassed all but one of the CSc 112 modules, this comparison is about the entire CSc 112 course, and not just the game practical. This is not ideal, but it was decided as necessary due to students experiencing survey fatigue, resulting in unreliable answers. The number of participants almost halved between surveys; an additional survey would have negatively affected this number even more severely. The comparison made between the two surveys is an indicator of how Rhodes University's introductory computer science course is affecting the attitudes and opinions of potential computer science students. This final survey also asked what the students' experience of the game was. These responses will be used to assess the success of the game. The students of CSc 112 were also observed as they played the game, to see whether more information could be gathered as to how problems were approached and what sort of questions were asked by students. This also helped verify or validate the responses to the surveys.

1.3 Research goals

The main goal of this research was to develop a game to teach computational thinking skills in contextualized way, and to access whether this would have a positive effect on how students perceive their own abilities and computer science in general. The research also aims to explore how this effect relates to females specifically, and to see whether contextualized learning through the use of games can combat the gender divide inherent in computer science. The research aimed to assess how the current CSc 112 curriculum was affecting the attitudes of students.

A subgoal of this research was to determine what has been done in previous research in regard to contextualized learning, especially with the focus on its effect on gender and being implemented through the use of games. It was also a subgoal to establish the current perceptions of contextualized learning and the lack of gender diversity in the field within the Rhodes University computer science department.

1.4 Research scope

The developed game only teaches introductory computational thinking skills to novice (or beginner) computer science students. Due to the multitude of external factors (such as previous experience, social interactions within the course, other modules and preconceived ideas of self and the subject) the exact effect of the game cannot be ascertained. The goal is see how students respond to learning contextually through the use of a game and to track changes in students' attitudes about themselves and the subject in order to see the effect CSc 112 may be having on its students.

1.5 Document structure

The structure of this thesis will follow the chronological process of the research. Chapter 2 explores previous work done on contextualized learning, teaching with games and the effect of gender in computer science. Chapter 3 discusses the design and results of two initial surveys run that informed the game development (see section 1.2 for an outline). Chapter 4 outlines the developed game's structure and how it was tested. Chapter 5 discusses the design and results of the final survey as well as the observations made during the use of the game by first year students. Chapter 6 presents the reached conclusions and ideas for future work.

Chapter 2

Related work

2.1 Introduction

University enrolment rates for computer science have dramatically decreased since 2000 (Bayliss, 2009, Carbonaro *et al.*, 2010, Carter, 2006), with some universities reporting that more than 50% of students that begin studying computer science abandon the field (Muratet, Torguet, Jessel, & Viallet, 2009). This has resulted in an abundance of literature that aims to re-evaluate how computer science is taught. Computer science should be presented as interesting, effective and minority inclusive (Margolis & Fisher, 2002). One of the main ideas towards meeting these criteria is contextualized learning, specifically through the use of educational games. Several universities and schools have begun using this approach and have received very positive feedback (Bayliss, 2009). Another issue to be addressed in computer science is the distinct male domination in the field (Margolis & Fisher, 2002). This literature review explores how teaching in a highly dynamic and contextualized way could increase the enthusiasm and understanding of students in computer science. This teaching style should also encourage female students to persevere and excel in the field, despite other factors that could be discouraging them. Some existing approaches to these problems are described and evaluated. There is also an exploration of what reasonings currently exist regarding the state of education within computer science. In particular, the subject of computational thinking is a focus area within introductory computer science.

2.2 Contextualized learning

There has been an increasing need in recent years for a more comprehensive and holistic presentation of science in education (Koul & Dana, 1997, Yager, 1996, Oers, 1998). It is no longer accurate to assume that teaching the same abstract concepts in every learning situation is effective (Oers, 1998). There is an underlying assumption in literature that context provides meaning to otherwise abstract concepts (Koul & Dana, 1997). It has been proposed that teaching computer science, and indeed any scientific subject, with a strong relation to real world contexts, increases student understanding and participation (Koul & Dana, 1997, Holman & Pilling, 2004). Oers (1998) suggests that using context in education is not just about teaching concretely, but turning a scientific concept into a relatable, everyday problem solving exercise. Owing to real world examples being highly complex and incorporating multiple ideas, explaining within a context allows multiple notions and experiences to be tied together. This allows concepts to form a more coherent whole (Oers, 1998). The information learnt no longer becomes an isolated formula or fact, but an integrated idea that has many applications and can be used in the decision making skills needed in day-to-day life. When learning is made abstract and static, students lose personal connection with the information, which limits motivation for learning (Koul & Dana, 1997). An important point made by Koul & Dana (1997) is that contextualizing learning does not make science subjective, but instead justifies it and gives meaning to theories. What is being taught can therefore be seen from the point of view where its value to life is just as important as the fact itself.

An example of contextualized teaching in action is a case study performed by Holman & Pilling (2004) where thermodynamics was taught by inserting contextual examples into the original course in an attempt to make the work more interesting and relevant. They received positive feedback about the level of interest and enjoyment in the course, but the added explanations made the course more time intensive, with some students struggling to apply the learned skills to different contexts. Perhaps this reveals a need to reassess what is being taught, not only how the material is being taught. The study was however only carried out over the period of only one year, and was compared with the teaching of a year's course by a different lecturer. The results are not conclusive but show encouraging progress while using contextualized teaching, with an average mark increase of over 10%.

With regard to computer science, contextualized learning can help bridge the gap between ordinary computer use and what is taught in computer science (Muratet *et al.*, 2009). The computer environments students interact with on a daily basis, to chat or to play, are very different from those they use for learning programming. The connection between these two environments is not immediately obvious to many students, and so by reinforcing the link between their everyday computer interaction and what they are learning could help prevent the theory of computer science appearing technical and tedious (Muratet *et al.*, 2009).

2.2.1 Attempting to restructure teaching practices

Koul & Dana (1997) fear that teaching in highly abstract and isolated ways results in a lack of engagement between the student and the material. They suggest that it promotes passivity and blind memorization amongst learners who are not required to interpret the course work for themselves (Koul & Dana, 1997). They even suggest the reason for this approach is to ensure control and discipline within a classroom as the students are not asked to engage (Koul & Dana, 1997). Their proposed solution was to restructure the way scientific subjects were taught. In their investigations they found that new concepts were introduced in a very structured order and their inclusion was not accounted for to the students (Koul & Dana, 1997). Students were aware of the way in which they would be tested on the concepts, of standard questions and answers, and so there was little to no incentive to explore the concepts further. The examples in textbooks were seen to be very limited and did not encourage further thought, or if they did, were too abstract or vague to engage the students. These limited examples used inductive inference to reach a static conclusion. Students would be shown a set of logical steps of how to solve a problem, which required almost no interpretation, merely acceptance that that is how that type of problem was solved (Koul & Dana, 1997). By teaching contextually, stale facts are seen in a real world application and the value of the material becomes more evident. This also means science becomes applicable at every level, instead of just at higher, more complex levels (Koul & Dana, 1997).

However, for a restructuring to occur within a curriculum, contextualized teaching has to be accepted by teachers and students. Geddis (1991) makes suggestions about how controversial issues can be evaluated within a classroom. He recommends looking at the intellectual independence of a student, evaluating how the student is capable of interpreting, speculating, judging and integrating ideas. By defining student leaning as more than their level of achievement, a deeper understanding of the process can be obtained.

2.2.2 Using contexts to personalize learning

Cordova & Lepper (1996) explored how contextualizing, personalizing and offering a choice in what was taught not only improved students' motivation, but also their level of engagement with the material. Owing to the positive outlooks, this approach also increased students' perceived level of competence and raised their aspirations. Cordova & Lepper (1996) commented on how the enthusiasm small children have towards learning is lost as they move through school, which they attribute to the decontextualization of instruction. The information being taught is no longer immediately relevant to them and presented in an abstract way that is meant to help generalize learning (Cordova & Lepper, 1996). By presenting material in a meaningful context, Cordova & Lepper (1996) intended to appeal to the intrinsic motivation for learning that is found in children.

Cordova & Lepper (1996) used computer games to teach mathematical concepts to fourth and fifth grade children. The first approach to try and encourage the students to be motivated about learning was to personalize several key features of the learning context to make the work appeal directly to students (Cordova & Lepper, 1996). The second strategy was to include an element of choice into the learning activities. This was done in an attempt to increase the student's sense of control and self-determination (Cordova & Lepper, 1996). When students are given a choice, they become more invested in the material they have selected, which has been shown to not only increase enjoyment, but also to make students perform better and be more persistent about completing tasks (Cordova & Lepper, 1996). Bayliss (2009) noted the same point, and also acknowledged that this level investment made students less likely to cheat.

The research Cordova & Lepper (1996) conducted used five groups of elementary school children, each of which was presented with different versions of a computer game that taught mathematics (Cordova & Lepper, 1996). Though mathematics is different from computer science, there is a strong correlation between success in mathematics and success in computer science (Carter, 2006). Half the students received games set in a generic fantasy setting, while the other half received personalized fantasy settings based on collected background information (Cordova & Lepper, 1996). Then half the students from each fantasy setting were given a game that offered a limited set of choices about features in the games, whereas the other half were not. There was also a control group which received the game with no fantasy element at all (Cordova & Lepper, 1996). This resulted in five groups: general fantasy and no choice, general fantasy and choice, personal fantasy and no choice, personal fantasy and choice, and the control. Of these groups, personal fantasy and choice showed the highest levels of enjoyment and were the most willing to extend their learning time after hours. Another finding of interest is that all the groups, except the control, were more than twice as likely to select a more difficult level than the control group, showing evidence of confidence amongst the students (Cordova & Lepper, 1996). Cordova & Lepper (1996) found no evidence that gender or race had any effect on the students' performance. This promotes the idea that contextualized learning encourages both genders equitably.

Koul & Dana (1997) also acknowledge the value of personalized contexts. They propose that how students problem solve is strongly influenced by social and cultural factors and that traditionally, abstract approaches to teaching scientific subjects ignore this fact. When teaching contextually, students can connect the work to their own prior experiences which should encourage students to articulate or find value in what is being taught (Koul & Dana, 1997). Again, this also involves the idea of choice. Students could interact with contexts that captured their interest. This both motivates learning and allows students to take on a more creative role in learning (Koul & Dana, 1997, Cordova & Lepper, 1996). Students can begin to take ownership over what they learn and "see themselves as producers of new knowledge" (Koul & Dana, 1997, p139).

2.2.3 Fantasy contexts in learning

Parker & Lepper (1992) developed a series of activities designed to teach school children basic programming logic. Their activities required the children to place themselves in the role of an arrow-like cursor that needed to navigate around the screen using provided commands. For the control group, the activities were presented in an abstract form, and in the other group, the activities were set in a fantasy scenario (Parker & Lepper, 1992). For example, the control group was presented with a screen with five circles on it and were asked to navigate around each circle so that they touched all the edges in turn, while in the second group the circles were made to look like islands and the students had to navigate around each edge to collect pirate treasure that had been buried there (Parker & Lepper, 1992). Though both activities require the same level of skill, the second group was far more motivated to participate and became more deeply involved in the tasks. After two weeks of activities, both groups received the same test and there was found to be a statistically significant difference between the two sets of students, with the fantasy context group outperforming the control group. By giving the activities context, the students in the fantasy group were more motivated to learn and more interested in what was being taught.

Casey, Erkut, Ceder, and Young (2008) attempted to use storytelling as a way to teach geometry more effectively to children. They based their study on previous researchers who had discovered that the story framework improved cognitive retention of material and information (Mishra, 2003). The use of storytelling also helped motivate students to learn (Cordova & Lepper, 1996). The findings of the study were that females benefited more from the contextualized teaching style than males.

Malone (1981) states that fantasy can either be extrinsic or intrinsic to game play. Extrinsic fantasy is external, with no impact on the actual game play, it is merely setting. Intrinsic fantasy is internal to the game experience and is potentially far more interesting and instructional than extrinsic fantasy (Dickey, 2006, Malone, 1981). Intrinsic fantasy could be used to indicate how a skill could be used in the real world or could provide analogies to aid understanding (Malone, 1981). The problems players will encounter in these imagined worlds will require them to actively and critically reflect on what they know to solve problems (Gee, 2003). There are benefits other than reflection to integrating a narrative into an education course, such as opportunities for evaluation, illustration, exemplification and exploration (Dickey, 2006). Narratives have also been shown to increase comprehension (Dickey, 2006).

2.2.4 Teaching computational thinking

Computational thinking is a fundamental skill required for programming. However, teaching computational thinking has different priorities to teaching computer science. It is more about a mode of thinking and a way of approaching problems than content. Essentially it is about abstraction (Wing, 2008). There is a strong emphasis in Lu & Fletcher (2009) to acknowledge the difference between programming and computational thinking. Lu & Fletcher (2009, p 260) outline four key points of computational thinking:

> 1) it is a way of solving problems and designing systems that draws on concepts fundamental to computer science; 2) it means creating and making use of different levels of abstraction, to understand and solve problems more effectively; 3) it means thinking algorithmically and with the ability to apply mathematical concepts to develop more efficient, fair, and secure solutions; and 4) it means understanding the consequences of scale, not only for reasons of efficiency but also for economic and social reasons.

What is important about these points is their universality. Computational thinking is not just for computer science but is a key skill in any subject that involves logical problem solving or information processing (Bundy, 2007). This is because it teaches students to think algorithmically, where they have a stepped procedure to take an input and produce some desired output (Wing, 2008). It encourages new kinds of questions, and using new approaches to reaching answers (Bundy, 2007). Also, finding enjoyment in computational thinking, a more basic skill than introductory level programming, is likely to encourage students to pursue computer science as a subject or interest.

2.2.5 Motivations for doing computer science

Carter (2006) conducted a survey to ascertain the reasons for students who show aptitude for computer science not choosing to major in the subject. As a result of the correlation between mathematics ability and computer science aptitude, over 800 mathematics students were interviewed from multiple high schools. She found that the various motivations for and against computer science were affected by gender.

The top reason for both male and female students not choosing to do a computer science major was the lack of desire to sit in front of a computer all day or that they had already decided to major in something else (Carter, 2006). This reiterates the positive effect that making computer science fun and interesting could have. A large percentage of students also wanted to choose a major that was more people oriented (Carter, 2006). The top male reason for selecting computer science was interest in computer games whereas the top female reason was to use what they would learn in another field. The third most popular reason was previous experience in the field. This last option was selected by significantly more males. Though females found more reasons to reject computer science than males, both genders had the same three top three reasons for taking computer science. It was found that 80% of the students surveyed did not know what was learnt in university level computer science. The top three reasons against computer science could be combated by educating students about the field itself. Teaching contextually could emphasize how computer science can be people oriented or be integrated into other fields (Carbonaro *et al.*, 2010, Carter, 2006). Computer science could potentially attract far more students, if those students knew what the field involved. By making computer science highly relevant, enrolment rates could dramatically increase.

2.3 Teaching with games

Games provide a dynamic and interesting platform from which to teach. A game can simulate a real world example and require real-time interpretations of concepts to solve problems and advance (Prensky, 2002, Dickey, 2006). The benefit of intermittent reward in games (such as completing a level or receiving a prize) helps motivate students to continue working (Cordova & Lepper, 1996, Prensky, 2002, Leutenegger & Edgington, 2007, Muratet *et al.*, 2009). If students have fun while using the application, they become encouraged to play more than the required amount (Chang *et al.*, 2012). Extended interest promotes learning as students are more likely to explore and look for new ways to apply their acquired knowledge (Prensky, 2002, Chang *et al.*, 2012). Also, contextualized cases require students to make practical decisions based on learnt principals, allowing them to learn concepts by working with them (Koul & Dana, 1997, Dickey, 2006). Another advantage of using games in education is it encourages collaborative learning (Muratet *et al.*, 2009).

Good games provide information on demand, in a situation where that information is needed, not out of context as information in schools often is (Gee, 2003). Information is introduced gradually as it is needed, instead of in a large chunk with an attempt at implementation afterwards. Also games can remain at the pace of the player, with more advanced players advancing quickly to a level where they are challenged (Gee, 2003). Education is often aimed at low level students so that no one is left behind, but this can often bore a lot of students (Gee, 2003). Allowing education to operate at the students' individual level of competence could counter act this.

Prensky (2002) praises the use of games for teaching because it allows the process of learning itself to motivate students. A game's main purpose is to entertain the player, which is why they are so engaging. If a similar mind-set is applied to teaching, students would become far more enthusiastic about participating in an activity that is actively trying to entertain them (Prensky, 2002, Dickey, 2006). Also, due to the visual aspect of games, students are able to almost immediately see the mistakes in their 'code' manifest (Leutenegger & Edgington, 2007). This also will help students visualize conceptually what their logic is doing.

2.3.1 Games and computer science

Leutenegger & Edgington (2007) used games to teach an entire introductory computer science course and found that this approach improved student understanding across all examined topics. This study greatly valued instant visual feedback for students to try to directly relate their understanding to what their code actually means (Leutenegger & Edgington, 2007). This approach not only increased understanding and retention, it also raised the level of enjoyment in the class. The course gained a new reputation for being fun and interesting. Having a better reputation doubled enrolment rates for the course and increased the number of students deciding to major in computer science.

Chang *et al.* (2012) found most available, modular games for teaching computer science to be boring and lacking in playability. They wished to use teaching materials which exhibited typical video game characteristics such as character development and skills improvement. In response Chang *et al.* (2012) developed the Dream Coders Project, which is a 2D role playing game that is actually a programming assignment framework. The game is complete and playable but is missing elements of functionality that can be filled in by students. Owing to the game functioning with no extensions, different functionalities can be added by students on different levels. Adaptability, creativity and generality of the game were key concerns, so that what was developed by Dream Coders could be used by educators in the future. Chang *et al.* (2012) used a quest-based role playing game as a structure, where specific programming concepts had to be implemented before each quest could be completed. The example they provide involved quests to locate a map that is locked in chest. To unlock the chest, the student must write a function to sort three random numbers (or pass codes). The main character of the game is a student who falls asleep in class and is then unable to wake up. To return to the waking world, he must complete the quests. The code for each quest is in a text based console.

When this game was presented to faculties for inclusion in computer science introductory courses, its battle theme and use of violence for advancement was questioned (many quests centred on defeating monsters). Owing to time constraints for the development of a game that covered all aspects of an introductory programming course, the end result was relatively simple, repetitive and sparsely populated. However, the idea is sound, and with more time, it could be developed into an effective teaching tool.

2.3.2 The advantages and disadvantages of using games to teach

Rochester Institute of Technology developed a program called Reality and Programming Together (RAPT), which used games as an application area to teach traditional computer science concepts (Bayliss, 2009). The retention rate of this course was a staggering 93%, compared to the university's regular computer science course, with a rate of 57% (Bayliss, 2009). Students also responded well to the course with regard to motivation and engagement, as 100% of students answering a survey within the RAPT course stated that the program should continue.

The RAPT program was begun in 2007, and two years later Bayliss (2009) wrote a summary on what had been learnt about using games to teach computer science. The first piece of advice offered to fellow educators was to remain focused on the course outcomes and not to become caught up in creating impressive graphical user interfaces. Initially, students became caught up in creating graphics for their games and ignoring the core algorithmic intent of their assignments. For the following class, graphics were provided so that students could concentrate on developing the problem solution. In the early stages of the course, the lecturers found it more effective to give the students partial solutions, and ask them to fill in small parts, than to ask them to build solutions from scratch (Bayliss, 2009). This let students get a feel for how solutions should look and taught them how to read other people's code/approaches. To assess if games were effective learning tools, pre- and post-tests were run to determine the students' progress. The results were very encouraging.

However, the RAPT course was not without problems. Bayliss (2009) also discussed the pitfalls and potential hazards of using games for education. In 2007 the introductory course used Wii remotes in co-ordination with an API and previously written code modules. A lot of problems were encountered with getting the technology to work correctly, and despite students thoroughly enjoying the remotes when they did work, they found them frustrating and distracting when they did not. Bayliss (2009) urges lecturers to consider how software or technologies could interfere with the concepts being taught. Another complication was the extra time required of the lecturer for developing a game course took over a traditional practical. This approach required more commitment from lecturers, which not all lecturers were willing or able to give. Bayliss (2009) recommended assessing whether the work and learning curve required for the game approach was indeed the best way to teach the course outcomes, and whether or not the current system is completely effective. The games used in university courses must definitely teach the required concepts, and not purely be used to engage students. However, if the course is extremely short Bayliss (2009) recommended valuing student engagement highly, as this would encourage students to continue working or exploring after the time period had elapsed. It is also important for a lecturer to be fully prepared to teach with a game. A poorly planned game assignment could be substantially less effective than a non-game assignment (Bayliss, 2009).

2.3.3 Developing computational thinking games

There are currently many examples of novice-programming environments for learning basic programming skills, such as Scratch, Alice2 and StarLogo The Next Generation (Muratet *et al.*, 2009). These all allow the exploration of computational ideas through developing basic 3D or 2D games or stories. Their basic visual languages allow code to be developed quickly, without students having to worry about syntax errors. However, the same variety of environments is not available for computational thinking as an isolated skill.

Repenning, Webb, & Ioannidou (2010) have been running extra-curricular activities on game design to try to motivate, engage and educate students interested in computer science. In an attempt to increase the reach of this program, they are trying to scale up the program and get it accepted into the required curriculum of public high schools in the United States of America. The project is called "Reforming IT Education through Game Design: Integrating Technology-Hub, Inner City, Rural and Remote Regions" (iDREAMS). To facilitate the program, they developed a checklist for educational computational thinking tools. The checklist consists of six conditions, all of which they feel are vital to effectively teach computational thinking concepts. Though these concepts were developed for use in high schools, they could be applicable for an introductory university course.

- **Low threshold**
  Students should be able to use and/or develop games quickly. If even simplistic games are excessively complicated, students will quickly become frustrated and lose confidence.
- **High ceiling**
  Though the games have to be simplistic enough for any student to use, scope for more advanced students must also be included. Students who are excessively enthusiastic or engaged should not be trapped in basic examples that limit their creativity or growth.
- **Scaffolding**
  The tools should provide stepping stones to ramp up skill and feed into more advanced programmes later on.

- **Enable transfer**
  It should be made obvious to students how the skills they learn from computational thinking tools can be applied to computer science and other science subjects.

- **Supports equity**
  The developed tools should be effective for motivating and educating students across ethnicities and genders.

Repenning *et al.* (2010) state that the important distinction between programming and computational thinking is that in computational thinking there is a simple and direct mapping between a problem and its solution. For example, they compare two teaching tools that use cursor-controlled characters: AgentSheets and Scratch. Both systems use drag and drop functionalities to build up basic algorithms. These simplistic, block-based graphical languages are used in many beginner programming environments as they all students to not worry about syntax and instead be solely focused on the problem solving (Muratet *et al.*, 2009). However, when more complex coding practices (such as doubly nested loops or abstract pixel offsets) are used, such as in Scratch, the visual language becomes far more similar to simplistic code than a basic problem solution. This is because they do not conceptually trace directly back to the original problem description (Repenning *et al.*, 2010). There should be a direct correlation between a solution and the problem description in computational thinking.

Muratet *et al.* (2009) wanted to develop a serious game that was targeted directly at students, while meeting computer science learning objectives. Upon researching what type of game to develop, they found strategy games to be the most popular game genre for both genders, with 57% of interviewed women in computer science playing these types of games and an even higher percentage for males. Strategy games usually have a virtual environment where resources are distributed across a map (Muratet *et al.*, 2009). They typically have three stages: harvesting resources, building structures and units and then fighting opponents. To win, a player must defeat the opposition or achieve some specified goal. There can also be sub-campaigns to teach concepts and all players to become adjusted to the environment. Good players need to plan ahead and react quickly. In order to use this genre for teaching computer science, there must be a system for inputting code that affects the game play. Two approaches were proposed: 1) enable in-game programming but limit the players control of characters to try and keep the flow of the game and 2) differentiate between coding and playing so they cannot happen simultaneously, rendering the player inactive during simulation of the written code (Muratet *et al.*, 2009).

2.3.4 Narrative design for effective games

The use of narrative is a way of framing problem-solving in daily experiences (Dickey, 2006). Narratives allow humans to give meaning to their experiences and knowledge. Thinking in a narrative framework allows experiences and concepts to be integrated into a plausible storyline (Robinson & Hawpe, 1986). Games, particularly adventure games, can have a strong narrative structure that could support problem-solving. Games can consist of goal-based scenarios that encourage development of skills based on content knowledge within a situational context (Dickey, 2006). In adventure games there are two main narrative techniques to motivate players: plot hooks and emotional proximity (Dickey, 2006). Plot hooks are a common literary technique to keep a reader/player engaged. They are unanswered questions that arouse the player's curiosity, and so they are compelled to try resolve them. The second technique makes the player empathize and connect with their character. To create this effect, similarities should be established between the player and the characters, through giving the characters characteristics players can identify with (Dickey, 2006). The character's motives within the narrative are also important, as this helps the player invest emotionally in the adventure (Dickey, 2006).

The setting and back story of a game is also important, as the player will make assumptions about what is plausible or available based on these factors (Dickey, 2006). There is a necessary balance between explicit and implicit clues to allow the player to reach their own conclusions without feeling abandoned or confused.

Dickey (2006) provides a framework for how to integrate the adventure game narrative into a learning environment. It consists of six categories:

- **Present the initial challenge**
  The core of any quest is a challenge. It will become the climax of the narrative and drive game play.

- **Identify potential obstacles and develop puzzles and resources**
  The journey to the main challenge should be littered with smaller obstacles and tasks that help develop the player's skill-set to solve the final problem. Throughout the journey through these sub problems, there should be resources, tools and tips to help the students overcome the challenges.

- **Identify and establish roles**
  Every other character the hero encounters through his journey will fulfil certain

roles, that each serve a particular function. The following are a few key character types:

  - The hero will be the embodiment of the student who is the agent of the action and should be interacted with in first person.
  - The mentor is a common archetype, a guiding figure to offer help and to drive the action forward.
  - The threshold guardian could be a character or a situation that tests the student's content knowledge. This guardian would have to be overcome using the gained skill set.

- **Establish the environment**
  Setting is an important element of the game play experience. It can be broken down into four key aspects:

  - The physical space in which the player moves around.
  - The temporal dimension of the game, or the role time plays. It includes aspects like how long a player has to complete a task or the time line the narrative follows.
  - Considerations such as whether the game is fantasy-based or realistic, historical or current, are important for contextualizing the narrative and help the player establish the game's boundaries.
  - The emotional and ethical background of the characters is important to define as this will reinforce the plausibility and realism of the characters.

- **Creating a back-story**
  The back-story should outline the environment as well as justify the main challenge.

- **Develop cut scenes to support the narrative story line**
  Cut scenes support the narrative throughout the game. They provide essential information and plot hooks to drive the action forward. They could also provide feedback on how a student completed a task.

All these elements work together to create an engaging narrative that can be used as a framework for educational exercises.

2.4 Gender in computer science

In today's world, technology and information have become more widely used than ever before. The acknowledgment of a digital divide occurring between economic classes due to lack of access to technology happened many years ago (Carter, 2006, Margolis & Fisher, 2002, Lau & Yuen, 2010, Horne, 2007). However, another digital divide exists between genders (Lau & Yuen, 2010). Females are under-represented in the field of computer science, and this may be due to a male-biased education system that does not adequately address female learning. The unpopularity of computer science amongst female students can be seen in Figure 2.1, where females choosing to major in computer science is considerably lower than in other major fields. Horne (2007) found that in schools, though there was no gender difference on standardized pen-and-paper tests, males performed better on computerized tests than females. This was attributed to a lack of confidence on computers amongst females.

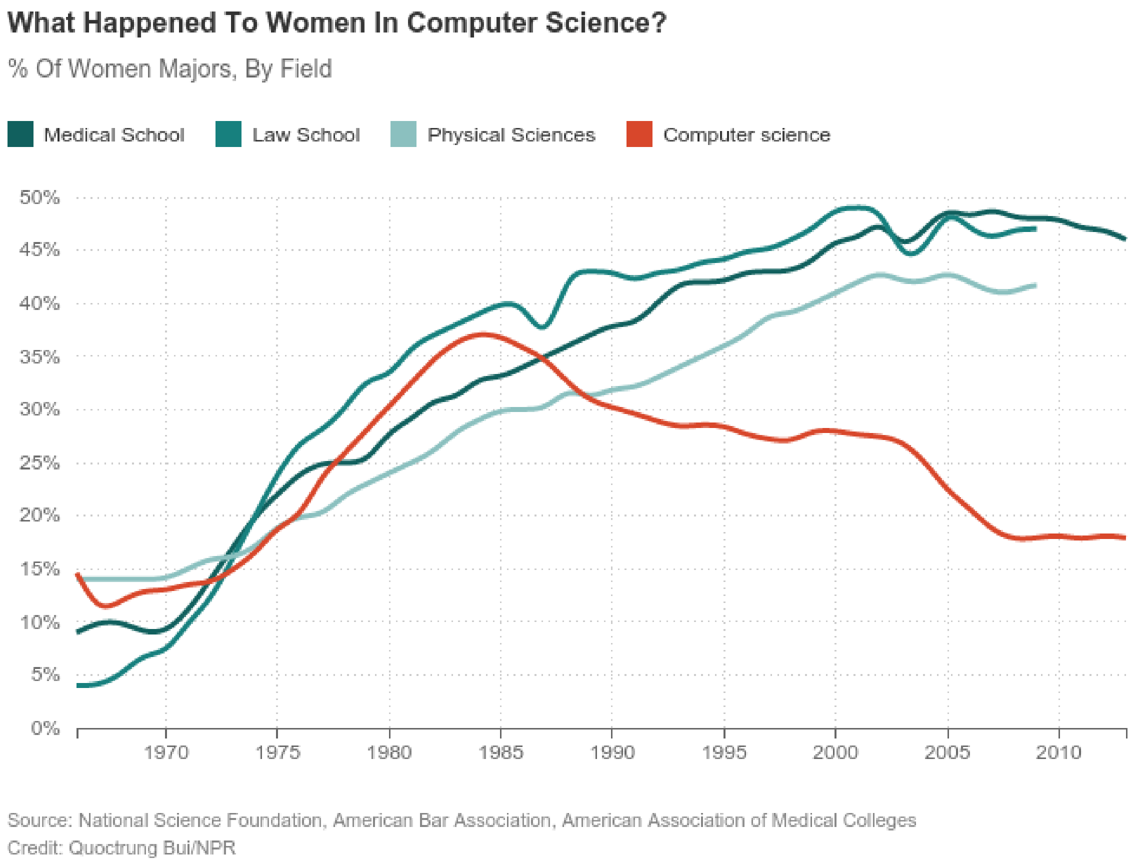

Figure 2.1: Graph of percentage of female majors by field from 1970 to 2010 in the United States of America (Henin, 2014)

In most countries, there are more females than males pursuing tertiary education, however the opposite proves true within computer science university courses (Anon., 2009). A review of scientific achievement in the United States of America (Anon., 2009) revealed that more females were enrolling in high school science courses than males. However, males were on average performing better academically in science subjects (Gunn *et al.*, 2002). Lack of achievement could lessen the enthusiasm in female students towards pursuing further scientific study. Clewell & Campbell (2002) have also suggested that stereotypes have played a role in the gender divide. Stereotypes are also further enhanced by the lack of female role models in science and often result in females receiving less encouragement to pursue a science-based career then an equivalent male student would receive (Anon., 2009).

In a camp run to encourage female high school students to become developers, it was found that contextualized and relevant projects resulted in the students feeling more secure and confident in the field. This leads to more of the females being enthusiastic about pursuing computer science at tertiary level than before (Burge, Gannod, Doyle, & Davis, 2013). However, contextualizing concepts has been proven to improve understanding and interest in both genders. There is an increasing number of such summer camps, after-school programs and computer clubs designed for female and minority students, particularly at high school level (Bayliss, 2009). This indicates that there is an increasing demand for students to be presented with such opportunities, and a wish from minorities to be specifically catered for within computer science.

There exists a danger of allowing the use of gender to result in an overly simplistic comparison of students, when many other factors, such as socio-economic or cultural factors, can affect academic performance (Gunn *et al.*, 2002). Ultimately, there should be an effort to redefine the discipline of computer science to be more gender-inclusive (Margolis & Fisher, 2002). Excluding females from the field not only results in an inequality, but it affects what is being produced by the industry. Margolis & Fisher (2002, p 3) make the observation that "females must be part of the design teams who are reshaping the world, if the reshaped world is to fit females as well as males". By attempting to create a more inclusive educational system and debunking the stereotypes about females in computer science, a more comprehensive and holistic future for computer science is possible.

2.4.1 Games and gender

Contrary to popular belief, females make up 45% of all game players, if casual gaming is included (Leutenegger & Edgington, 2007, Muratet *et al.*, 2009). As such the use of games is relatively gender neutral, if the content of the game is kept neutral (Carbonaro *et al.*, 2010). Females have been shown to be less attracted to violent and online multi-player games, however this is not true for all games (Leutenegger & Edgington, 2007). In fact, females can be equally motivated by learning through games as males (Leutenegger & Edgington, 2007). Repenning *et al.* (2010) reported a significant increase in female participation in computer science related electives after the inclusion of game design courses in high school curriculums. Rochester Institute of Technology developed a new degree for game design, which received 14% more female freshman than the traditional computer science degree program (Bayliss, 2009).

Carbonaro *et al.* (2010) found that using computer games to teach computer science to high school students showed equal success in both genders. Males did not dominate any of the measured outcomes of the study (higher-order thinking, computer science abstraction skills and activity enjoyment).

Laurel (1998) lead a team of researchers to try to discover why females were being left behind by the rapidly advancing gaming industry. They concluded that the type of adventure game that would appeal to females would feature a 'real-life' setting as well as new places to explore. Females preferred games with story lines and a leading character they could identify with (who could be their friend). Friendship was seen as an important aspect of the game. Females require feeling social and safe in the gaming environment, and prefer being able to design, create and communicate (Margolis & Fisher, 2002). Males, however, preferred games with violent feedback, such as ending the game by the main character dying or by killing another character (Margolis & Fisher, 2002). Males also strongly favoured adventure-style games that had a level of escapism (Margolis & Fisher, 2002). Another genre that appeals to both genders is strategy games, where logic and planning allow players to move ahead (Muratet *et al.*, 2009). This both appeals to a sense of safety in females and males' wish for escapism (especially if the game is centred around war strategies).

Jorgensen, Logan & Lowrie (2013) explored how computer games can be used for contextualized learning by trying to break away from the traditional structure of educational games. Instead of using a drill-and-practice framework, they aimed to mix education and entertainment to use a narrative-like framework with a more informal approach to learning. This would also have the added benefit of the learning not being perceived as a 'lesson' to students. This study also focused on the difference in responses and abilities between genders in game play. The skill Jorgensen *et al.* (2013) were concentrating on was map reading and interpretation. In their initial surveys they found that though males were more likely to play map-based games, females were more likely to play problem solving games.

The method used by Jorgensen *et al.* (2013) was to have a male and female student from early high school play a map based game and then interview each player throughout their progression through the game over three weeks. They were interviewed about how they were playing and how they felt about the game. One student from each gender is too small of a sample size to have trustworthy results but their results are interesting to remark upon. The female student was far more deliberate in her approach to the game, less likely to use trial-and-error than the male student. It was also noted that the male relied heavily on graphical cues whereas the female student relied more on hints and written information provided in the game. The male learnt through doing and exploring, whereas the female waited to be taught. It was noted that the male student had far more experience playing games than the female, and so would have a higher level of confidence than the female. A difference observed by Tartre (1990) in sixth-grade students was that students who had strong verbal skills but low spatial visualization skills focused on verbal clues for their solutions to math problems and had the lowest mathematics scores. Females tend to have stronger verbal skills than males, and so are less likely to follow visual clues or translate verbal information into pictorial form (Tartre, 1990).

An important suggestion made by Jorgensen *et al.* (2013) was that games could be used to trigger an awareness of real-world settings or contexts where different skills and approaches could be applied. Dickey (2006) made a slightly different point that narrative games provide environments that allow students to practice and gain skills that have a use in the real world. This second observation disconnects the game narrative from a real world application, but if a skill can be learnt in a way where its application is understood, it should be easier to transfer across contexts and uses. In other words, giving an indirect context, whether in a fantasy or real world example, enhances the concept.

2.4.2 Gender and self-efficacy

Beyer *et al.* (2003) examined the gender difference amongst computer science majors and non-majors with regard to a number of different attributes to assess the sort of people fulfilling both roles, and how they perceived themselves as well as each other. They attribute the gender divide in computer science to two factors: negative stereotypes regarding the field and low levels of confidence. They found female computer science majors to have a lower level of confidence on a computer than male non-majors. However, female computer science majors on average found CS classes less overwhelming than male computer science majors. This indicates a false level of self-belief. They also found that though males and females valued the subject equally, males had higher aspirations within the field. Quantitatively, there was no gender difference in ability. The difference lay entirely in self perceptions, confidence and motivation. Establishing a sense of enjoyment and confidence for females in computing is an important aspect of narrowing the gender gap (Margolis & Fisher, 2002).

In mathematics, there is a much higher percentage of male university students achieving good marks than females (Bandalos, Yates, & Thorndike-Christ, 1995). However, it must be acknowledged that males have a much wider distribution across mark ranges than females (Casey, Nuttall, & Pezaris, 1997). Bandalos *et al.* (1995) did a study to see the relation between test anxiety and achievement. They observed a difference between genders. Self-efficacy was found to have a slightly stronger relationship to test anxiety or worry for females than males, meaning females are far more affected by the level of their achievement than males. Casey *et al.* (1997) found there to be a strong relation between gender and anxiety, with females exhibiting higher levels of math anxiety than males, but these anxiety levels did not directly relate the marks achieved by either gender. Bandalos *et al.* (1995) also found that students were more anxious about being evaluated if their previous experience in the field was less than average or had been negative. As females often have had less experience in computer science than females (Margolis & Fisher, 2002), this could affect females more than males. Casey *et al.* (1997) found males to be significantly more confident in their abilities than females, which is consistent with the other study. Bandalos *et al.* (1995) found that females who attributed success to external factors had a much lower level of mathematical skill than females who attributed success to their own efforts. Males who blamed failure on external failure had the lowest level of stress.

2.4.3 Gender and learning styles

Due to the high level of male participation in science subjects, it has been suggested that the actual course work and assessments have been developed with male bias (Clewell &

Campbell, 2002). The early stages of computer science courses are usually focused on technical aspects of programming, with the development of multipurpose and useful systems being left until the later years of the degree, resulting in the coursework appearing to be entirely removed from any real world-context (Margolis & Fisher, 2002). Rosser (1990), a feminist educator, believes that ensuring that science and technology courses are considered within their social context is of paramount importance within education. She states that openly discussing the benefit of the course work with respect to the environment and other people is advantageous to both genders. This approach contextualizes the information and raises the level of perceived importance. It has been shown that females tend to perform better with open-ended or essay type questions while males perform better when tested with multiple choice or short questions (Clewell & Campbell, 2002). If computer science is taught in smaller, dissociated chunks, it encourages male-type aptitude more than the female. A computer science professor, Dianne Martin, suggested that an integrated approach to computer science, with greater value placed on the social impact and relevance of computer science fundamentals, would help redress the balance between genders (Margolis & Fisher, 2002).

Shaw & Marlow (1999) found that though there was no obvious difference between the genders with regard to learning style, there was a significant difference in the level of comfort with using computers. Males felt much more at ease with new concepts and technology whereas females resisted moving away from what they were already comfortable with (Shaw & Marlow, 1999, Chamillard & Karolick, 1999).

2.5 Chapter summary

Enrolment rates in university computer science courses are dropping around the world (Bayliss, 2009, Muratet *et al.*, 2009) and literature suggests making computer science more fun and relevant for students could be an effective solution for counteracting this problem (Bayliss, 2009, Carter, 2006, Casey *et al.*, 1997). There have been a great many successes when using contextualized learning to teach science subjects (Casey *et al.*, 2008, Cordova & Lepper, 1996, Holman & Pilling, 2004, Koul & Dana, 1997, Parker & Lepper, 1992). It appears to be an effective teaching tool that aims to entertain, motivate and actively engage students in learning. Games in particular provide effective, interactive environments to test student knowledge within a context that could be related back to real life (Bayliss, 2009, Cordova & Lepper, 1996, Jorgensen *et al.*, 2013, Muratet *et al.*, 2009, Prensky, 2002). Teaching contextually through games also appears to equally encourage both genders, proving to be a very inclusive approach to education (Bayliss, 2009, Casey *et al.*, 2008, Leutenegger & Edgington, 2007). Female students in particular have low confidence levels in the field, which could be raised through motivational and supportive teaching practices (Bayliss, 2009, Margolis & Fisher, 2002).

The difference between computer science and computational thinking has been acknowledged by several authors (Lu & Fletcher, 2009, Repenning *et al.*, 2010, Wing, 2008), yet almost all computer-science-related educational games have been designed to teach programming basics (Bayliss, 2009, Cordova & Lepper, 1996, Leutenegger & Edgington, 2007). There exists a space in education for a game which develops basic computational thinking skills. These skills would be useful to more than just computer science students. They could benefit any student doing a subject that involves logical problem solving or information processing (Lu & Fletcher, 2009).

The next chapter outlines the design, results and analysis of the surveys run amongst the Rhodes University computer science department and the CSc 112 students.

Chapter 3

Initial surveys

3.1 Inter-departmental survey

A survey was run throughout the entire computer science and information systems departments at Rhodes University, from second year up to lecturers. The purpose of this study was to assess how contextualized learning was perceived within the field and what opinions were held on the gender divide within the department. The information systems department was included to increase the number of possible participants and because CSc 112 is a prerequisite course for information systems.

3.1.1 Participants

The survey had 61 participants, of which only 4 were lecturers. The gender distribution was skewed towards males as expected (33%, n=20 female; 67%, n = 41 male). The number of participants from each department was fairly even, with some participants falling into both departments (57%, n=35 computer science; 59%, n=36 information systems). However, the gender distribution from each department was not even (Females: 25%, n=5 computer science; 70%, n=14 information systems; 5%, n=1 both; Males: 49%, n=20 computer science; 29%, n=12 information systems; 22%, n=9 both). This means only 10% of participants doing computer science were female. This correlates to the statistics from the United States of America where 12% of computer science degrees are awarded to females (Zweben, 2012).

3.1.2 Method and analysis

The survey was run from 6 to 13 May 2014. Ethical approval for the survey was sought from and awarded by the Hamilton Ethics Board (application tracking number: CS14-02). Participants were made aware of the intentions and purpose of the study, and that their participation was voluntary and confidential. There were three questions to ascertain the demographics of the participants and then seven subsequent questions on the topic. Each question allowed participants to submit any additional comments they might have.

The survey was conducted electronically using Google Forms and the collected data was stored anonymously in a spreadsheet and as an automatically generated summary. Only basic descriptive statistics were gathered from the survey responses.

3.1.3 Results and discussion

The results of each question are discussed below. All the questions asked within the survey are listed below as headings, with the results and the analysis of the results listed below the relevant heading. A complete list of the questions can be found in Appendix B[1].

Do you feel the relevance of Computer Science and computational thinking is portrayed in CS lectures?

13% of participants felt the relevance of computer science and computational thinking was always made apparent in computer science lectures, 44% felt it was usually explained, with the rest of the group stating that it was either occasionally apparent or not at all. It is interesting to note that not a single female said that relevance was always explained. Though 52% of female participants did think it was usually explained, this is far lower than the 61.5% of male participants who felt it was always or usually explained. The fact that only half the females in the department feel that computer science is being portrayed as relevent is an indicator that more of an effort should be made to place what is taught in context. There were a few comments that suggest that the relevance of computer science is obvious and does not require explaining. These were all male comments.

[1] All responses to the questions asked in the interdepartmental survey can be found in electronic format on the accompanying CD-ROM.

Do you think there is a lack of computational thinking and problem solving skills amongst first year students?

Of all participants, 8% felt there was an extreme lack of computational thinking and basic problem solving skills, 46% said that most students did not have the skills, 28% said that only a few students struggle and 7% said the skill level was really good, with the rest of the participants selecting that they did not know. These percentages were fairly consistent across genders (however, only males said there was an extreme lack of skills). Comments centred around students having no previous experience from high school. With more than half of responses indicating that there is a lack of skills, computational thinking abilities need to be directly addressed.

There is evidence to suggest teaching concepts while concurrently explaining real world contexts and applications of the data increases retention and interest in teaching material. Do you feel this approach is used within the teaching of computer science, problem solving or computational thinking?

With this question about whether computer science is taught contextually, the distribution of responses between the genders was also the same. 21% said they definitely felt this approach was definitely used, 39% selected that it was mostly used, 34% said it was used a little, and finally 3% said not at all. This is encouraging as more than half the participants said they felt that computer science at Rhodes University was taking a contextualized approach.

From the comments it was clear that participants did not however fully understand the question. They did not understand how technical core concepts could be related to anything concrete. This is where the fundamentals of this project lie, injecting real world importance into the very fundamentals of the subject and changing the focus from pure concepts to application. A specific comment that should be addressed is “Students should be better able (and we should demand) that they learn to join their own dots. If we persist with a philosophy that tries to spell everything out in a spoon-feeding kind of way, it weakens the eventual education”. This comment was made by a male lecturer. This is exactly the opposite of what is being attempted. Purely conceptual teaching presents dots with little to no material on how to connect them. Contextualized learning creates the dots, and the connections between them, so that the relationship between dots can be understood more fully. As discussed in section 2.2.1, the attempt of contextualized learning is to see the value concepts have, and to expand them, not constrain them.

On the other end of the spectrum was a comment made by a male student studying both information systems and computer science: “Real world contexts are not used often enough but they are still used. There is this worry in the back of my mind that what I am learning now may well be completely outdated by the time the real world comes around. To reassure me that this is not all doom and gloom, it would be helpful to experience more real life contexts when teaching computer science, problem solving or computational thinking.” This is one of the key concepts of contextualized learning, to reassure students that what they are learning is relevant, applicable and deeply connected to the world around them.

What do you enjoy most about programming and development? Rate the following with 5 being very enjoyable and 1 being the least enjoyable.

The participants were asked to rate the following based on how much they enjoy them: Problem solving, Satisfaction of success, Creativity, Control, Challenge, Ownership and Logic. Again there was almost no difference between male and female participants. Both genders selected ‘Satisfaction of success’ as the most enjoyable aspect of computer science. A game should appeal to this type of enjoyment as games reward success with congratulations and awarded points. This was made a specific focus for the game.

The second most popular element for females was Problem solving and for males it was Creativity, but by a very small margin. The game designed in the project was centred around problem solving (the third most popular element for males) and so should appeal greatly to this element. However, there will be limited creativity due to the simplistic nature of the game made necessary due to time constraints. Creativity should stem more from programming than computational thinking problems, and as the game does not actually deal with programming, this limitation should be acceptable.

Science is often said to be taught in a highly abstracted and theoretical way, with little emphasis on real world application and favouring shorter questions without needing explanations. Do you think that the way computer science is taught is in this style?

The learning style described by the question is a traditionally male way of learning (Clewell & Campbell, 2002). To avoid responses being tainted by the idea of male learning when asked whether computer science was taught in this modes, the question was phrased indirectly. In this question there was finally a difference in response distribution:

Table 3.1: Results of question inquiring about computer science's bias towards male learning

	Average (n= 60)	Male (n= 40)	Female (n=20)
No, not at all	8%	10%	5 %
Perhaps slightly	56%	67.5%	35%
It is fairly biased	23%	22.5%	25 %
Yes, it is very biased	7%	0%	20 %
I don't know	5%	0%	15 %

It is interesting to note that not a single male respondent felt the field was biased yet a fifth of females did. This indicates that how the participants are viewing the subject from a learning perspective is different between genders.

Do you think changing the way Computer Science is taught would affect the difference in gender representation within the field?

Of all answers, 30% of participants said they did not know if changing the way computer science is taught would have any effect on the gender diversity in the field, 44% said it would have no effect and 25% said yes it would. Of this 53% of female participants said no. Comments indicated that participants felt that the divide was symptomatic of society, not of the way CS is taught. It was also suggested that university level was too late to make a change, that cultural norms and beliefs are already concreted by this stage. Many comments showed the misconceptions and gender bias ideas that exist in the field. There were also a few male comments about not understanding why the gender imbalance is a problem. However, there were a few insightful comments: "Why would there be a gender difference in different disciplines if not for the way it is taught and applied?" (male student). Another comment discussed the effect that choices of examples used in class can have (linking back to relevance). If examples of concepts are skewed towards a specific gender, it could affect how different genders perceive the field.

Do you have any final comments about contextualized learning or the gender imbalance within the field of computer science?

The responses to this question been separated into comments about contextualized learning and comments on gender within computer science. These groupings are discussed below.

Contextualized learning comments

"Computer Science is a practical subject that shouldn't be taught in just a classroom where the lecturer demonstrates the codes to the students and they just observe the code, it should rather be taught in a computer lab where students don't only observe demonstrations from the lecturers but also do the coding themselves and really see how things work. This would open up an environment where students interact more with lecturer and engage with the lecture material without only having to wait for the prac day to ask questions."

This comment by a male student describes an interactive class environment where the focus is on *how* to use what is being taught, instead of *what* is being taught, which is the main goal of contextualized learning. Here it is being requested by a student, which is encouraging as to how this approach to education would be received by students. The next comment reveals that some students have the opposite opinion on the subject.

"Contextualized learning is not highly emphasised in the CS courses that I have taken. It seems that the view is more of giving students tools with which to tackle any context in which these tools can be relevant. The advantage of that is that if these tools are fully grasped then the student has the ability to apply them to a wide variety of settings, whereas if they are taught in a particular context, there is a risk of associating the tool with the context. The disadvantage is that most people learn best based on something they have seen before or can relate to, so a lot of people fail to grasp the fairly abstract theories. I vote for the former."

This is a common fear amongst participants. This male student understands the value of pure abstract concepts and the short falls that badly implemented contextualized learning can have. However, contextualized learning should not remove the use of abstract concepts, merely change the focus of how they are taught so that they are easier to understand, and can possibly be more deeply understood. Placing value on the information (by showing its use) can also increase student retention (Bayliss, 2009, Mishra, 2003).

Which leads onto this comment, made by a male lecturer: "Contextualized learning can be used to complement traditional learning, but should not replace it". This is arguably the best place to start, as it would encourage lecturers and students to not react against the new learning structure but would also allow an expansion of learning practices. The developed game is used along side the traditional lectures, so students still receive the benefits of both techniques.

Gender related comments

"Hah! So many. Like the stupid comments that men say to women in pracs – you can't do a problem? Must be 'cos you're a woman'. Though a man having the same problem doesn't have the same flak directed at him. Or the comments that women receive from other women, 'oh you're doing CS? that's so nerdy!'. Or the social interactions online... brogrammer culture is both retarded and retarding the entire field. Or the general attitude among some students that science is "hard", so women shouldn't do it, it's not 'feminine'. So it's isolating to do CS, here and in the real world and in many places online (though some online places, e.g. StackOverflow, are quite good). If I was a woman, I'd think it over a few times before taking a swim in these shark-infested waters."

This comment was made by a male lecturer and highlights the cultural and social issues that surround gender in computer science. Though contextualized learning cannot combat this directly, if more females could be encouraged to stay, cultural norms would adjust in response. If more female high school students saw computer science as a viable career path that many women choose, perhaps more females would persevere through the field. Another comment, this time by a male student, is also an indicator that the gender issue is understood within the field:

"I personally believe that the gender imbalance is a very real problem. I say it is a problem, because in my belief, Computer Science is another artistic medium– albeit a medium based in logic– that as a field would benefit greatly from differing perspectives. At the moment, society seems to shun females who delve into the realm of Computer Science, and I believe this should be changed. Perhaps, if Computer Science was advertised better– a lecture or two pertaining to potential avenues that Computer Science is involved in– the fairer sex would be more enticed to join in on the magic. Another dissuasive factor could be that Computer Scientists engender personae that are not indicative of all spheres that the field encompasses; you do not need to like games or watch my little pony to be a Computer Scientist, in fact differing perspectives would, in my opinion, propel the field forward in a myriad of amazing avenues."

Apart from stereotypes and gender norms as mentioned in the above comment, another reason for the gender divide is different learning styles. A female student that does computer science made the following observation:

"Our brains function differently. Men are naturally more left brained (on the whole) and girls more right brained (also a generalization). I don't think it has to do with the fact that we are male and female, but is completely dependent on interest and brain structure. If you notice, few of the girls that survive CSC are the ones that you could imagine being typically girly. They too are the more left brained of the bunch. So its left brain vs right brain, not male vs female."

Though this could be true, many left brained females are choosing not to study computer science. However, this comment reveals the importance of looking into a variety of elements that affect the number of females within computer science.

Despite the many insightful comments, some (male) students do not acknowledge the gender difference as an issue: "I think it it is fairly balanced probably due to the fact that I haven't considered it yet as it has not been brought to my mind until now" and "I don't think the gender imbalance is a problem because the examples etc are not aimed towards a particular gender. Therefore both genders have the freewill choice as to whether to take computer science or not". This research brings awareness of the issues of the gender divide within computer science. Participants in the surveys will now be thinking about and questioning the diversity amongst their classmates and peers.

3.1.4 Summary

Contextualized learning received mixed responses from the participants. A meeting of ideas between contextualized learning and traditional teaching seems to be the most accepted approach as many fear that purely contextualized learning will cause students to gain a shallow understanding of the concepts being taught.

Many of the asked questions received similar answers from both genders. This could indicate that the females that remain in the field see the field as males do, which is why they have remained. One of the few questions that indicated a difference in opinions was the question asking about male learning being used in computer science. This indicates a necessity to explore more gender neutral teaching practices, such as contextualized learning.

3.2 Pre-intervention survey

A survey was run in July 2014 at the beginning of the CSc 112 academic course to assess the students' attitudes towards computer science. This was done in order to confirm the research found in prior works as well as to provide a baseline to identify any changes in the students' perceptions of computer science and their own self efficacy. The questions posed to the students were based on the Computer Science Attitude Survey by Wiebe & Miller (2003). Responses were given on a five point Likert scale that ranged from "Strongly agree" to "Strongly disagree". The survey consisted of 24 questions which covered five areas of the students' attitudes towards computer science:

- confidence in their own abilities
- their attitude towards success
- how they perceived the gender dynamic in the subject
- the usefulness of computer science
- their motivation for taking the subject

These same questions were asked to the students again near the end of the course.

3.2.1 Participants

The survey was run amongst the students enrolled in CSc 112 at Rhodes University. 405 students participated, of which almost half were female (50%, n=203 female, 50%, n=202 male). Of this group only 18% were considering majoring in computer science (7%, n=26 females; 11%, n=45 males).

3.2.2 Method and analysis

The survey was run during July 2014 in an introductory practical. A complete list of questions has been included in Appendix C[2]. Ethical approval was sought from and approved by the Hamilton Ethics Board (application tracking number: CS14-06). Students

[2] All responses to the questions asked in the pre-intervention survey can be found in electronic format on the accompanying CD-ROM.

were made aware of the intentions and purpose of the study, and that their participation was voluntary and confidential. The survey was conducted electronically using Google Forms and the collected data was stored anonymously in a spreadsheet. The results were statistically analyzed using Statistica. The response Likert scales were converted to ordinal numbers (with "Strongly agree" being given the highest value of 5 and "Strongly disagree" the lowest value of 1).

The information to be gathered by this survey was to investigate the responses of students in regard to their own opinions and attitude, and specifically if these attitudes differ between gender. All responses were analysed using basic descriptive statistics in order to gain an understanding of the overall attitudes felt in regard to each question. A one-way ANOVA with assumed equal variance was performed in order to test the null hypothesis that there was no difference between the responses of both genders. A factor analysis was also performed to ensure that the questions were grouped by concept as expected by Wiebe & Miller (2003).

3.2.3 Results and discussion

3.2.3.1 Descriptive statistics of responses

One of the first questions participants were asked was why they were taking the course. They could select more than one answer from a combination of reasons.

Table 3.2: Results of why participants enroled in CSc 112

	Percentage (n= 438)	Female (n=223)	Male (n=215)
You have to take it for your degree	76%	78%	74%
You want to learn more about how to use a computer	39%	41%	36%
You would like to learn basic programming skills	31%	31%	31%
It seemed like an easy credit	6%	4%	7%
A friend told you it was a fun course	3%	2%	4%
You do not feel confident on a computer and want to learn more	17%	17%	18%
It sounded interesting	26%	31%	22%

The low values of 3% that indicated they had heard the course was fun and 26% that thought the course sounded interesting shows that CSc 112 does not have a good reputation amongst students. Games could help change how the course is perceived, and therefore increase student retention and enrolment (Bayliss, 2009). The only noticeably large difference between gender responses in this question was "It sounded interesting", where 9% more females chose this. This indicates that there is scope for increasing the number of females that remain interested in the field.

The next 24 questions required the students to rate their responses to statements from "Strongly agree" to "Strongly disagree". For simplicity, some of the questions will have "Strongly agree" and "Agree, but with reservations" grouped into agreeing with the statement, and vice versa for disagreeing.

When asked about to rate how likely the participant was to major in computer science, 45% selected "Strongly disagree" (50% female; 39% male) and only 8% selected "Strongly agree" (4% female; 13% male). The uneven gender distribution in the field is again confirmed by the fact that three times as many male participants have decided to study computer science than female participants.

Only 11% of participants strongly agreed with the statement "I have a lot of experience with computers" (4% female; 18% male) and 30% agreed but had reservations (28% female; 32% male). With an average of 33% of participants remaining neutral to the question (40% female, 26% male), a much higher percentage of females were neutral than males. This could indicate that females are undervaluing the amount of experience they have due to a lack of confidence, as only "Strongly agree" and "Neutral" have a noticeable difference between genders (Horne, 2007).

The majority of participants (53%) either agreed with "Generally I feel confident on computers". However, this was an average response that consisted of 62% of male responses and only 43% of female responses. Again, a much higher percentage of females were neutral to the statement than males (35% female; 26% male). This could again be attributed to females having a false sense of their abilities or of females traditionally being less encouraged to use computers (Margolis & Fisher, 2002). 21% of female participants disagreed or strongly disagreed and only 11% of males gave these responses.

"Strongly disagree" was chosen by 47% of participants in response to "I already have programming experience" (55% female; 38% male). Only 9% said the "Strongly agree" (4% female; 14% male) and a further 9% said "Agree, but with reservations" (7% female; 12% male). The expected gender distribution is again present.

Half the participants, 51%, were inclined to find programming interesting (44% female, 59% male). The number of females potentially interested (with 16% selecting "Strongly agree") shows that there are female students who could be potentially pursue computer science as an interesting subject, however only 4% are strongly considering majoring in the subject. This means 12% of strongly interested females are choosing to not pursue something they find interesting. A third of participants were neutral to programming (35% female; 26% male) and almost a fifth were not interested (21% female, 14% male).

The responses to the statement "Programming looks very difficult" received almost identical answers from both genders (with no more than 2-3% difference between genders). Half the participants agreed with the statement (52%), "Neutral" received 29% of responses, and finally 19% of participants disagreed. The significance of similar and differing responses between genders is discussed in Section 3.2.3.2.

As with the previous statement, "I think I can learn programming" received almost identical answers from both genders. 52% strongly agreed, 44% agreed, 3% disagreed and 1% strongly disagreed. That 52% of females strongly think they can learn programming affirms what was said above, that there are a lot more females who could potentially pursue computer science as a major but are choosing not to.

"When I find a problem difficult, I usually just give up" also had very similar answers from both genders. A huge majority of participants (89%) disagreed and only 11% agreed. That most females claimed they did not give up on problems they did not understand goes against what was found by Beyer *et al.* (2003). However, many females may view the failure in this context as giving up, instead of not solving the problem. Therefore, the developed game will still address the need for students being comfortable with not being able to solve a problem immediately, both through creating levels specifically for this problem as well as offering constant encouragement to motivate the student to continue trying.

The same majority (89%) agreed with "I enjoy solving problems", which is encouraging as a main feature of computer science is problem solving.

The need for achievement in the subject was obvious, as 98% agreed with "I really want to excel in this subject". This means students are motivated to work hard at the subject.

Clewell & Campbell (2002) pointed out the harm stereotypes can have on female representation in the field, however are stereotypes about studying computer science also prevalent? The fact that 83% of participants disagreed with "People would think I was a nerd if I did well in programming" would indicate that this is not a large problem amongst the students of CSc 112. This is reiterated in the next statement "If I did well in this subject, I would prefer no one knew" where 84% disagreed.

The disbelief of female related computer science stereotypes also showed positive results. In response to the statement "Females are as good as males at programming", 70% of participants agreed (76% female; 65% male). 24% were neutral to the statement (19% female; 28% male), which could indicate that the stereotype that females are worse at programming is believed more than the stereotypes about computer science being nerdy, particularly by males.

However, support of this stereotype does not extend to "Studying programming is just as appropriate for women as for men" where 83% of participants agreed (89% female, 77% male). Again there is a noticeable difference between the responses of genders. Almost an even number of females were disagreeing with the above gender related subjects as males, though this number is small (5-7%), it still indicates that some students believe the stereotypes that support the lack of female representation. A similar trend is noticed in the responses for "It is hard to believe a female would do as well as males in programming", with 76% disagreeing (80% female; 71% male) and 9% agreeing (7% female; 11% male).

However, the responses for "It makes sense that there are more men than women in computer science" are less definitive, with 49% disagreeing (55% female; 42% male), 29% remaining neutral (26% female; 33% male) and 22% agreeing (18% female; 25% male). This is not necessarily related to stereotypes. This statement could be affected by the acknowledgment of stereotypes. So the lack of gender diversity may not be supported, but the higher number of male computer scientists is understood within its social context.

Slightly more than two thirds of participants agreed that "Programming is a very important skill to have" and a quarter remained neutral. The value of computer science is understood by the majority (with females and males giving equal responses). On a broader level, 92% of participants thought that "Problem solving is a very important skill to have". As computational thinking is a collection of formalized problem solving techniques, this shows many participants value the skills that computational thinking would teach them.

Three quarters of the participants agreed with the statement "Computer science is a worth while and necessary subject", with 20% remaining neutral. A large majority of participants see computer science as a relevant subject but a smaller majority see programming as having relevance to themselves, as 60% of participants disagreed with "Programming has no relevance to my life" (60% female; 60% male). As relevance is one of the key areas contextualized learning hopes to address, this indicates that the relevance of the field as a whole is already obvious to most students. However, the inter-departmental survey revealed that actual computer science lectures were making the relevance of individual topics or concepts obvious to half of females and two thirds of males (see Section 3.1.3). These numbers match the responses to "Computer science is a worth while and necessary subject". It is possible that the students seeing the relevance of computer science in the department always saw computer science's relevance.

Despite being an introductory course, 86% of participants disagreed with the statement "Taking this course is going to be a waste of my time". This means that most students hope to learn more about computers and feel that it will be a good use of their time. Again this reiterates that computer skills are seen as relevant and important.

"Programming is boring" was included to see whether this could be a reason so few females are showing a keen interest in computer science, however 55% of participants (and 55% of females) disagreed with the statement. 36% were neutral and 9% agreed. That a small percentage found programming uninteresting means there should be another reason so few students are uninterested in pursuing computer science as a major.

To assess if students were avoiding computers in general, not just computer science, the statement "I try and avoid computers as much as I can" was posed. 82% disagreed (78% female; 85% male), 10% were neutral (13% female; 6% male) and 9% agreed (9% female; 8% male). Males were more opposed to the statement (with 69% selecting "Strongly disagree") but still a large majority of females felt the same way. This indicates that it is probably an indication of the subject that there are a smaller number of enrolments, not the high amount of computer use in the subject.

The final statement, "I am very excited to learn more about computers", got slightly different results from each gender. As can be seen in Table 3.3, a third of females were neutral about learning more about computers which is double the number of males who feel the same way. This could indicate that many females do not find computers exciting, and therefore have no desire to pursue an interest in them.

Table 3.3: Descriptive statistics results for "I am very excited to learn more about computers"

Response	Average (n= 436)	Female (n=222)	Male (n=214)
Strongly agree	47%	45%	49%
Agree, but with reservations	25%	20%	31%
Neutral	23%	31%	15%
Disagree, but with reservations	2.5%	3%	2%
Strongly disagree	2%	1%	2%

In conclusion, the trend found in literature that females tend to have lower confidence levels about their abilities in regard to computer science (Beyer *et al.*, 2003) was confirmed in the survey. Both genders were found to be enthusiastic towards learning computers and programming, an encouraging result. The stereotype that achievement in computer science was "nerdy" was not believed and so this is not a likely cause for the declined enrolment rates. However, males and females disagreed about whether it was appropriate for females to do computer science. Computer science, programming and problem solving were seen as relevant skills that were important to learn. Few students already had programming experience.

3.2.3.2 Investigating difference in the responses between genders

Due to the size of the survey, only certain results that proved significant in the one-way ANOVA will be discussed here (survey results in Table 3.4). The first question showed there is a statistically significant disparity between male and female participation in the subject (p-value<0.05). There was a statistically significant difference between the number of males and females that are planning to continue with the subject. There was also a significant difference in the questions related to confidence levels and interest in studying the subject (questions 2, 3, 4, 9), with males proving consistently more comfortable on computers as anticipated by prior research (Casey *et al.*, 1997). Using games to teach could create a fun and encouraging environment for females to build their confidence. Establishing a sense of enjoyment and confidence for females in computing is an important aspect of narrowing the gender gap within computer science (Margolis & Fisher, 2002).

An interesting response is that there is a large disparity between genders' beliefs about whether programming is interesting (question 5). With females entering university with

Table 3.4: Initial CSc 112 Survey one-way ANOVA results gender comparison

	Question	N	Mean	F	p
1	I plan to major in computer science	404	2.155941	8.4095	**0.00394**
2	I have a lot of experience with computers	402	3.191542	13.72	**0.00024**
3	Generally I feel confident on computers	403	3.501241	24.93	**0.00000**
4	I already have programming experience	403	2.166253	20.886	**0.00001**
5	Programming looks interesting	402	3.472637	15.469	**0.00010**
6	Programming looks very difficult	403	3.498759	0.17135	0.67914
7	I think I can learn programming	403	4.210918	0.00209	0.96360
8	When I find a problem difficult, I usually just give up	403	2.027295	0.88701	0.34686
9	I enjoy solving problems	404	3.779703	7.251	**0.00738**
10	I really want to excel in this subject	402	4.477612	2.6339	0.10539
11	People would think I was a nerd if I did well in programming	404	2.146040	0.26779	0.60510
12	If I did well in this subject I would prefer that no one knew	404	2.004950	0.14596	0.70263
13	Females are as good as males at programming	405	4.167901	8.6169	**0.00352**
14	Studying programming is just as appropriate for women as for men	404	4.443069	7.7912	**0.00550**
15	It is hard to believe a female would do as well as males in programming	404	1.710396	6.7295	**0.00983**
16	It makes sense that there are more men than women in computer science	404	2.490099	5.8203	**0.01629**
17	Programming is a very important skill to have	400	3.942500	0.29554	0.58700
18	Problem solving is a very important skill to have	402	4.639303	0.0305	0.86145
19	Computer science is a worth while and necessary subject	403	4.089330	0.19791	0.65665
20	Programming has no relevance to my life	404	2.247525	0.00407	0.94914
21	Taking this course is going to be a waste of my time	401	1.581047	1.587	0.20849
22	Programming is boring	403	2.235732	0.26185	0.60914
23	I try and avoid computers as much as I can	402	1.659204	3.8589	0.05170
24	I am very excited to learn more about computers	403	4.111663	3.5823	0.05912

already solidified ideas of their opinions on computer science, future works should look at encouraging females from high school, or even primary school. Both genders agreed that programming appeared difficult (question 6) but they also both agreed that they could learn the skill (question 7). This is encouraging as it indicates that both female and male students (over 90% of the sample) believe they are capable of learning programming. If the students believe they could learn programming, why are more students not exploring this option? The answer may lie in question 9, 'I enjoy problem solving' (p-value<0.05). Problem solving is one of the key features of computer science. Though many females do enjoy problem solving, males enjoyed it more by a significant factor.

All the questions relating to gender within the survey have been shown in Table 3.4. In every question, there is a significant p-value for the difference in responses between genders. Females were consistently more supportive of their gender in the field, and males less so. A level of bias towards females in computer science is endemic. Many females can become discouraged by this and chose to work in more female friendly environments (Margolis & Fisher, 2002). If more females were encouraged to remain within the field, and were able to succeed, the reputation of the gender could change. Clewell & Campbell (2002) believe that stereotypes are one of the main causes of the gender divide within computer science.

Both genders acknowledged the importance and relevance of programming and problem solving (questions 17 - 19 with p-values>0.05). Perceived importance of educational subjects helps motivate students as they are invested in the topic (Muratet *et al.*, 2009).

3.2.3.3 Factor analysis

In Table 3.5 the results of a factor analysis are shown, with values larger than 0.3 considered significant. It is interesting to note that the factor analysis loadings have different factor groupings than those identified by Wiebe & Miller (2003). Factor 1 shows a relationship between people who are planning to major in computer science with the people who enjoy the subject, have previous experience, value the subject and want to excel. This is to be expected as these students will already have high value placed in the subject due to their decision to select this major. What is interesting is the relationship this factor has with question 13, "Females are as good as males at programming". This is a positive discovery as it means students choosing to study computer science are supportive of female students who chose to programme, however none of the other gender related

Table 3.5: Initial CSc 112 Survey factor analysis (Questions with no factor values of significance have been removed)

	Question	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
21	Taking this course is going to be a waste of my time		0.33875			
1	I plan to major in computer science	0.51487	0.30718			
2	I have a lot of experience with computers	0.35619	0.73347			
3	Generally I feel confident on computers	0.47644	0.67915			
4	I already have programming experience	0.41377	0.63433			
5	Programming looks interesting	0.67965				
7	I think I can learn programming	0.62908				
9	I enjoy solving problems	0.49653				
17	Programming is a very important skill to have	0.37396				
24	I am very excited to learn more about computers	0.65709				
19	Computer science is a worth while and necessary subject	0.64468				
18	Problem solving is a very important skill to have	0.35150			0.36352	
10	I really want to excel in this subject	0.43120			0.31489	
11	People would think I was a nerd if I did well in programming				0.41037	
20	Programming has no relevance to my life				0.38005	
22	Programming is boring				0.32931	
14	Studying programming is just as appropriate for women as for men			0.55150	0.49179	
13	Females are as good as males at programming	0.40230		0.51545		
8	When I find a problem difficult, I usually just give up					0.43777
6	Programming looks very difficult					0.50786
16	It makes sense that there are more men than women in computer science					0.33009
	Expl var	5.27328	2.37291	2.16122	1.42852	1.26171
	Prp. total	0.21972	0.09887	0.09005	0.05952	0.05257

questions have been grouped. All other female related questions pertain to females entering the field, as opposed to the females who are already in the field being good at programming. The result of this is that female computer scientists are acknowledged but new students are not being encouraged.

Factor 2 in Table 3.5 groups the students that indicated that they had previous experience and computer confidence with those who felt an introductory course would be a waste of their time. This is logical as these students would feel they would not be learning anything new in an introductory computer science course. A high ceiling game could help combat this, as students would feel entertained and would be able to perform at a higher level than the course requires. The game developed for this project cannot fully meet this requirement, but this information provides validity for future development.

Factor 3 relates two gender related questions, but not all 4. It make sense that questions 15 and 16 are not related as they are negative comments on female participation. It must be noted that the factor analysis was redone with the negatively phrased questions inverted (strongly agree became strongly disagree and so on) in order to ensure that this was not affecting the groupings, but it made no difference to the classification of the loadings into factors.

Factor 4 has contradictory results. It groups students that want to excel and think that programming is important with students who feel the opposite, as well as including a gender related question. This may be due to the threshold value being set too low. If the threshold was adjusted to 0.4, this factor would group students who responded similarly to computer science being viewed as nerdy and who think it is appropriate for females to study computer science. This is interesting as they are both related to stereotypes.

Factor 5 interestingly relates responses to giving up on difficult problems with people who think programming appears difficult. This is a logical connection as people who immediately view programming as difficult will be less motivated to try. What is interesting is the grouping of a negative gender divide question in Factor 5. Perhaps this implies that students unmotivated to try in computer science are more readily disregarding female participation due to their assumptions of the subjects' difficulty.

The loadings of the factor analysis were roughly grouped into the initial five categories mentioned in Section 3.2, but not exactly. The validity of the survey has already been proven by Wiebe & Miller (2003), and so this indicates that this group behaved differently to their test group. However, the groupings of the loadings has provided new insights and confirms some assumptions made (such as those who already have experience will find an introductory course a waste of time).

3.2.4 Summary

Most students enrol in CSc 112 for course requirements of their other subjects. Only about a third are taking it to learn how to use a computer, to learn programming skills or because the course sounded interesting. Students are not choosing to enrol in CSc 112 because they heard it was a fun course. However, many students were excited to learn about computers and did not think that computer science achievement was "nerdy". Males believed stereotypes about computer science being inappropriate for women far more than females did. Both genders saw computer science, programming and problem solving as important and relevant skills.

Less than half the questions about the students' attitudes disproved the null hypothesis that there would be no difference between the responses of each gender. The ones that did produce a significant p-value (<0.05) were related to confidence levels, interests and the gender divide in computer science. All of these elements confirm what was found in previous research.

The factor analysis showed that the survey questions did not correspond exactly with Wiebe & Miller (2003). Factor 1 and 2 did correlate, however Factor 3 (related to gender) did not contain all gender related questions, only positively phrased ones. Factor 4 gave contradictory results and did not correlate to a factor described by Wiebe & Miller (2003), but is related to the effect of stereotypes in computer science. Factor 5 is relates students that have preconceived ideas about the difficulty of computer science with students who do not try hard to solve problems.

3.3 Chapter summary

This chapter describes the two initial surveys used to assess the opinions of the Rhodes University computer science department and the students enroled in CSc 112. It was found that there were mixed responses within the department to contextualized learning but most participants were open to the concept. Most questions received similar responses from both genders, however it was identified that the genders viewed whether computer science teaching was biased towards traditional male learning differently, indicating that it was not, in fact, gender neutral.

The pre-intervention survey given to the CSc 112 students revealed computer science and problem solving skills were seen as relevant and important. There was a difference in the self efficacy levels of each gender, with females being on average less confident in their abilities. Both genders were motivated to achieve in the subject.

The next chapter describes how the game was implemented and the structure of the game.

Chapter 4

Game implementation

4.1 Design decisions

The game was built and designed using GameMaker Studio, Professional edition, version 1.2. This was the section choice for game development platform. The original choice was Unity (developed by Microsoft) which was far more powerful but was to difficult to learn in the available time constraints. GameMaker was chosen as an alternative due to its ease of use and as a professional version of it was already available to develop in. For a relatively simplistic 2D game, such as the one developed, GameMaker was more than adequate.

The game was compiled as an HTML5 webpage and was run off the provided Rhodes University honours student research website. This platform was chosen over having an executable due to the ease with which it could be made available to the students. The images and sprites used in the game were either retrieved from free sources or created from scratch. The game only runs in Firefox and Internet Explorer browsers, with Google Chrome having HTML5 APIs that do not support GameMaker Studio.

An executable version of the game can be found at www.cs.ru.ac.za/research/g11z1718/game/[1].

[1] The GameMaker Studio files can be found in electronic form on the accompanying CD-ROM.

4.2 Game structure

The genre of adventure style game was chosen due to having a strong narrative framework that allows goal based scenarios that encourage skill development to be integrated into a situational context (Dickey, 2006). Due to the time constraints of the research, a fully realised narrative could not be developed. However, strong narrative themes run throughout the game. To set up the storyline, there was an initial animated scene with conversation between the main character, to be controlled by the player, and the teacher, who acts as the antagonist in the game (as seen in Figure 4.2). The cutscene ran as an animation that students had to watch; it could not be skipped. The scene sets up the motivation and goal for the game while establishing the roles of the characters. This helps increase player investment in the game (Dickey, 2006). The focus of the game was placed on the problem solving challenges, rather than the narrative, allowing the game to remain casual. The motivation for this choice is that casual gaming appeals to a greater percentage of both genders than very complex gaming (Leutenegger & Edgington, 2007, Muratet *et al.*, 2009).

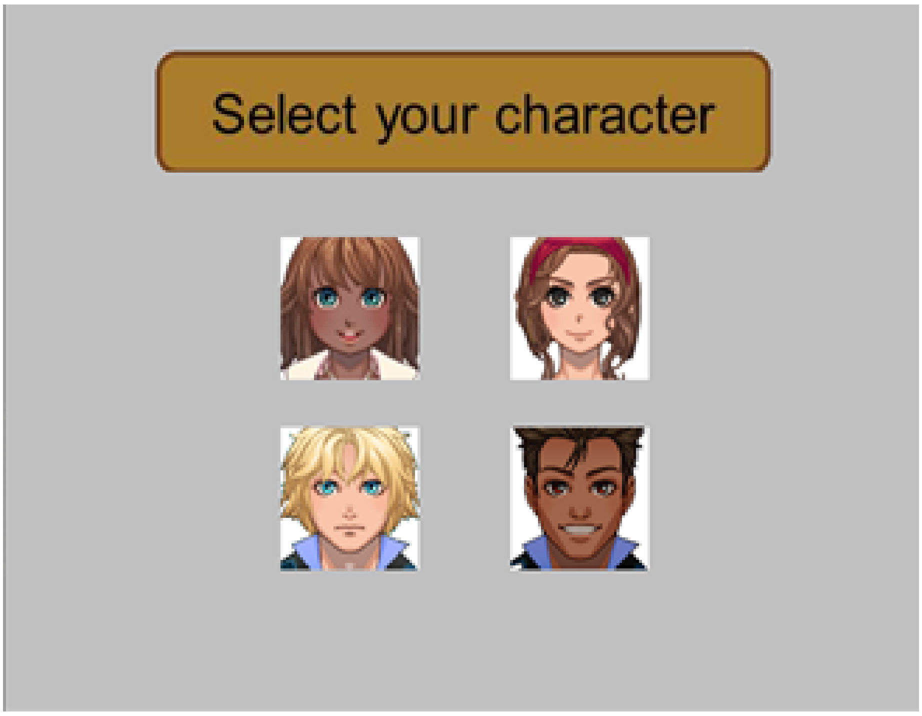

Figure 4.1: Character selection screen

To include an element of personalization in the game, players were allowed to a select their character from four choices, as shown in Figure 4.1, which included a variety of genders and ethnicities. This allowed as many players as possible to select a character that resembles themselves (showing self identification within the game). Personalization acts to improve students' motivation and their level of engagement with the material (Cordova & Lepper, 1996).

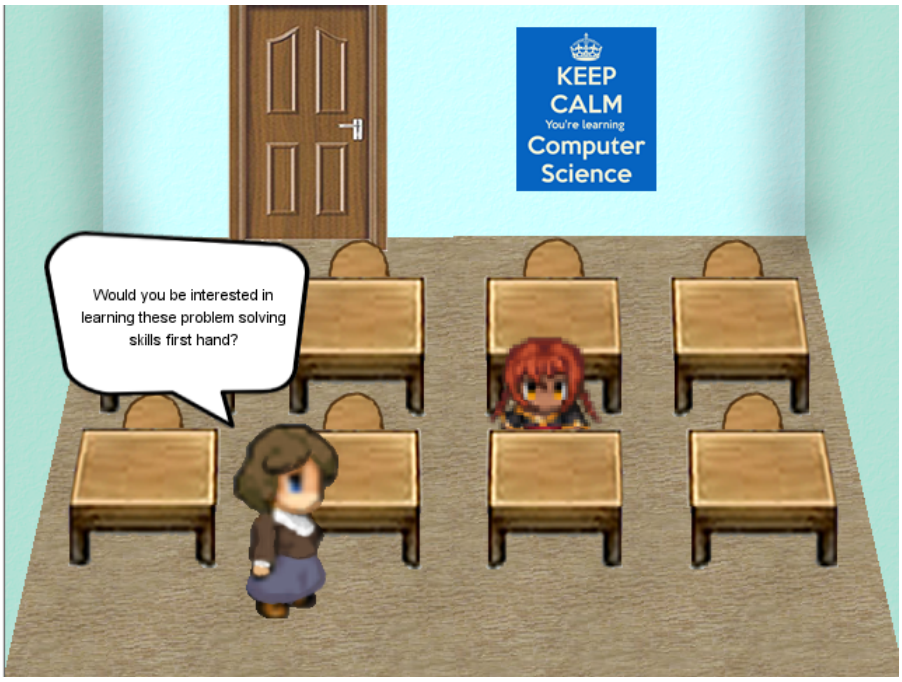

Figure 4.2: Narrative cutscene

The narrative concept was that the student has stayed after class to speak to their teacher about how long the computer science course was going to last, as they are getting bored in lectures. The teacher suggests an alternative solution, going on an adventure to learn the problem solving skills first hand. The student readily agrees and follows the teacher into her office. A second cutscene begins (as seen in Figure 4.3) where the teacher wishes the student luck and a secret panel opens in the wall, revealing an elevator which the student enters. They emerge underground in a series of passage ways filled with doors containing each section (Figure 4.4). A single speech bubble explains what happened and that the player can use the arrow keys to move around in the passage ways. This narrative element gives the player a sense of purpose for the game and places the challenges the player faces in context.

Throughout the game, players are presented with this map so they can track their progress. Each section is represented by a door on the map. The challenges in the game were grouped by concept: trial and error, if statements, variables (and abstraction), while

Figure 4.3: Second narrative cutscene

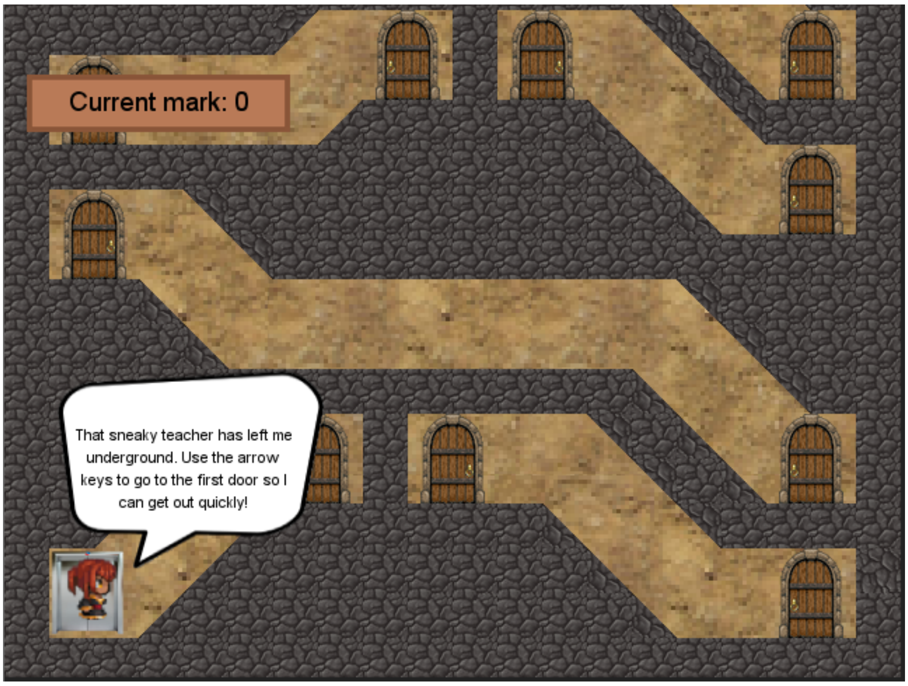

Figure 4.4: Map of sections and introductory commentary

loops and algorithmic thinking (and pattern detection). The trial and error section was selected due to the difficulty many students, especially female students, have with failure (Beyer *et al.*, 2003). In order to encourage students to simply try different approaches and be unafraid of failure, this section has no neat and obvious solutions, requiring students to simply try. the other sections were selected based on the module requirements for CSc 112 and the computational thinking skills typically taught in the module.

The player is required to advance through these sections in order as the sections build on each other. Each section begins with the teacher giving an explanation of the concept (Figure 4.7). The player can read these instructions at their own pace, using arrows located at the bottom of the screen, and can choose to skip through them very quickly if they already understand the concept. This was allowed to maintain a high ceiling where students who have already learnt the concepts or would rather learn by experimentation can do so without becoming frustrated (Repenning *et al.*, 2010). Also, slow readers, or players that rely heavily on written instruction, can get the full benefit of these explanations. Jorgensen *et al.* (2013) found that male students relied heavily on graphical cues, whereas female students rely more on hints and written information. If the player fails to succeed in the upcoming levels, they are offered the choice to return to the instructions. After finishing each level, the player receives encouragement and congratulations, trying to use positive reinforcement to build the player's confidence. They also receive marks based on their performance in the level. The player can track their progress through each section through a title bar in the top right corner that states the section and the level number.

The first introductory level is entirely mouse based, comprising of a mixture of logical and trial-and-error style problems. These were included to encourage the students to make use of what they had without receiving any instructions. As previously discussed, female students tend to be hesitant to act without hints or clear directions (Jorgensen *et al.*, 2013). Problem solving does not always come with these hints, and so this section aims to encourage students to branch out from needing to be told what to do. It also an attempt to build their problem solving confidence as they can solve the levels on their own, without needing hints. The instruction room for this section welcomes the player to the dungeons of the school and explains that they need to solve all the problems to work their way back to the surface. They are told that the next few rooms will offer them no instructions, and they must simply preservere through the challenges.

The first level (Figure 4.5) required the player to figure out that what was required of them was a colour code. This was discovered by clicking around the room and seeing

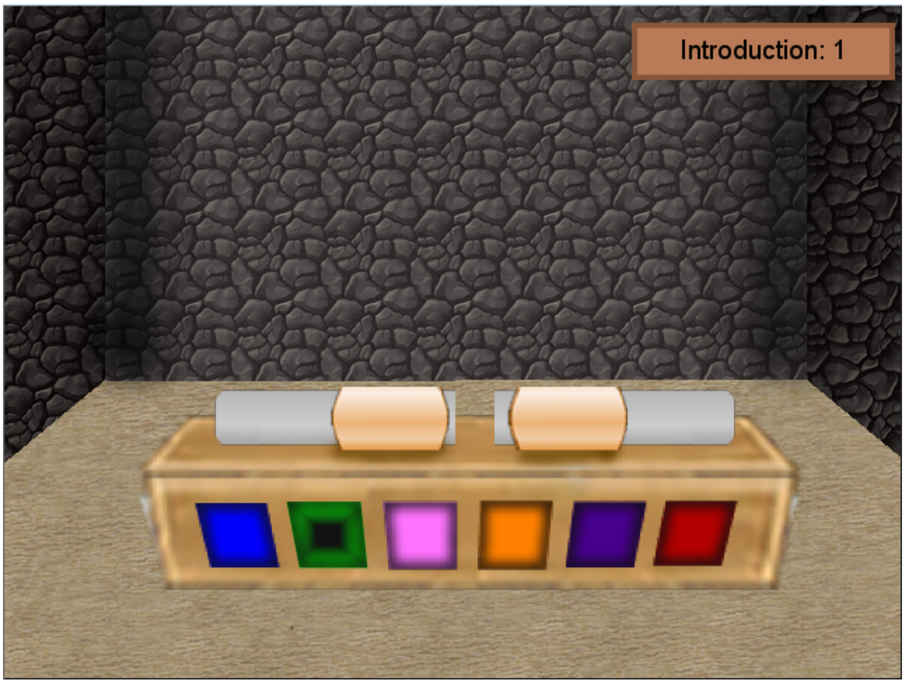

Figure 4.5: Introductory trial and error game

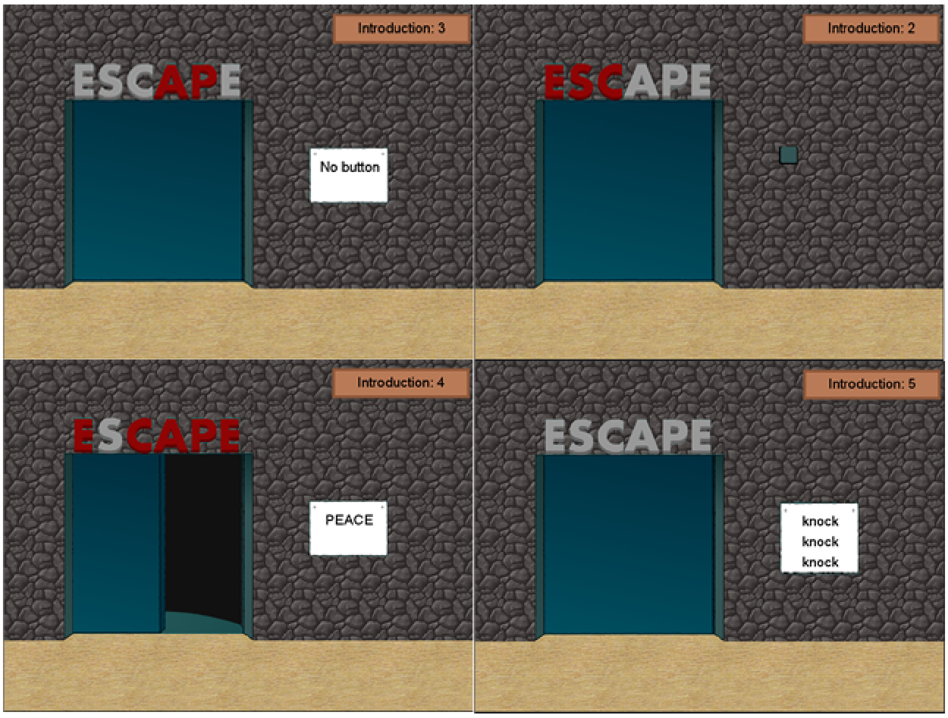

Figure 4.6: Escape games to use situational problem solving

that there was a visual response to pressing a certain colour panel, and further response if they pressed a specific second panel, and so on. The next four levels are basic escape games that come with little or no instructions on how to get the door to open (Figure 4.6). These problems require lateral thinking and the trail and error problem solving approach. After each level the player is told how many marks they earned from playing and congratulated on their efforts, in order to encourage them to continue and to remain motivated.

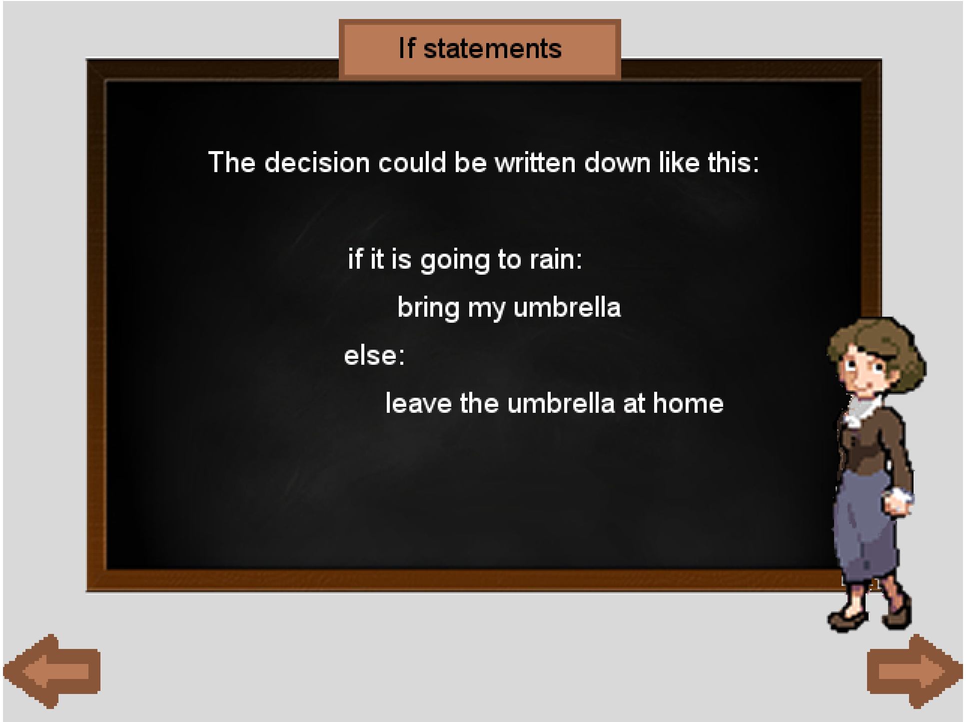

Figure 4.7: An excerpt from the if statements instruction room

The next section demonstrated if statements. This set of levels required the player to sort items into buckets using a set of if statements as an indicator of how to sort them (Figure 4.8). The levels increase in difficulty, beginning with just one if statement and then increasing to several cases and nested if statements. The instructions for this section were split into two parts so that the player could try simple statements and understand them, before receiving an explanation for how multiple if statements can interact. This allows the focus to remain on the context of the information instead of inundating the player with information before they can explore what they have been taught. After each item is sorted, using either the number keys or the mouse, a large cross or tick appeared above the item for a second so the player received feedback on their selection.

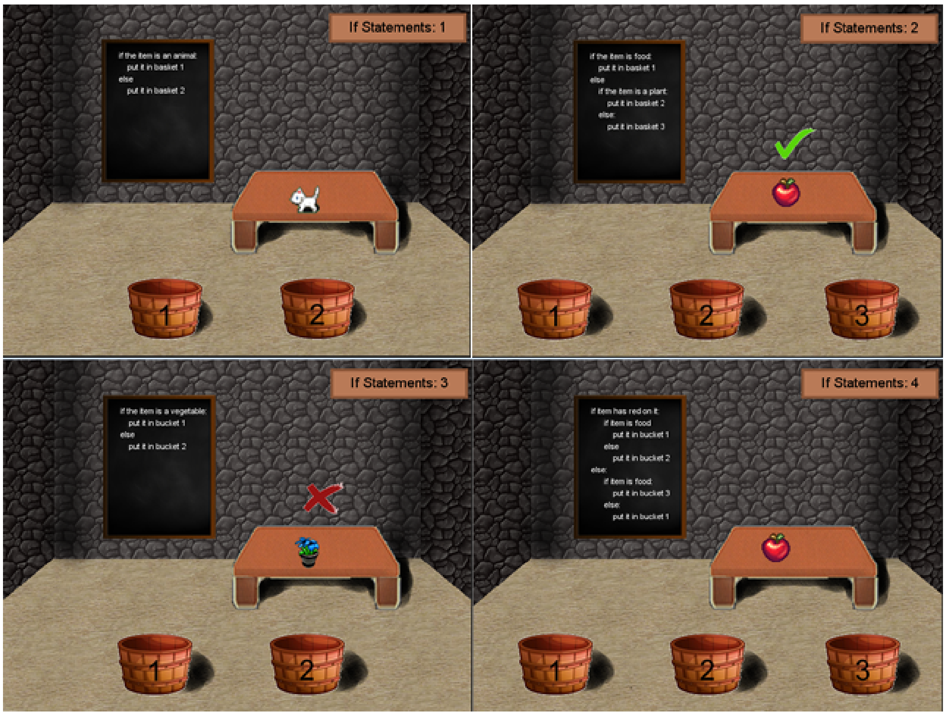

Figure 4.8: Sorting games to demonstrate if statements

The variable section instructions focused on how the use of variables allows for abstraction, and why abstraction can be so powerful and useful. The instructions used the previous sorting rooms as an example of how a variable could be used. This was done in order to allow the student to begin to connect concepts together, so they are not seen in isolation or as existing specific to one situation. For this section, the players had to track expressions and give the value of the variables at various points within the expression (Figure 4.9). One level also required the students to place four expressions in order so that the variables were assigned before they were used in calculations.

As with the if statement section, players were presented with while loop expressions that they had to follow. The narrative structure of the game was that there are rooms filled with booby-trapped tiles and the player had to follow the while loop expressions exactly, or they would not make it across the room. If the player makes an incorrect move, an explosion occurs and the player must restart. This allows some violent feedback that males respond well to, but within a problem solving framework that females enjoy (Margolis & Fisher, 2002). The early levels only had one while loop, and then one while loop containing if statements (to show that different structures can interact), until finally there were nested while loops (Figure 4.10).

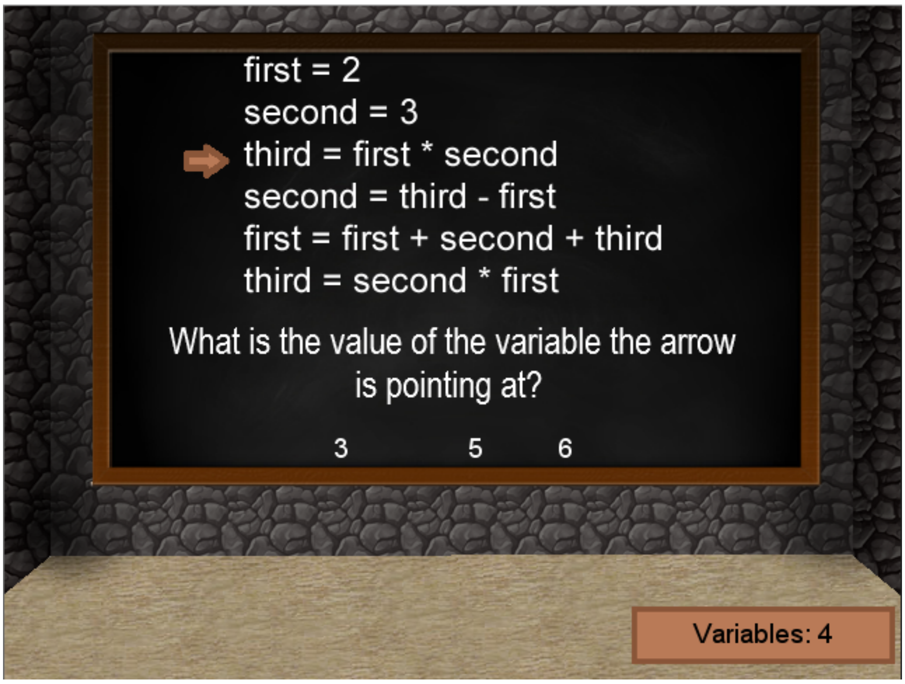

Figure 4.9: A variables level

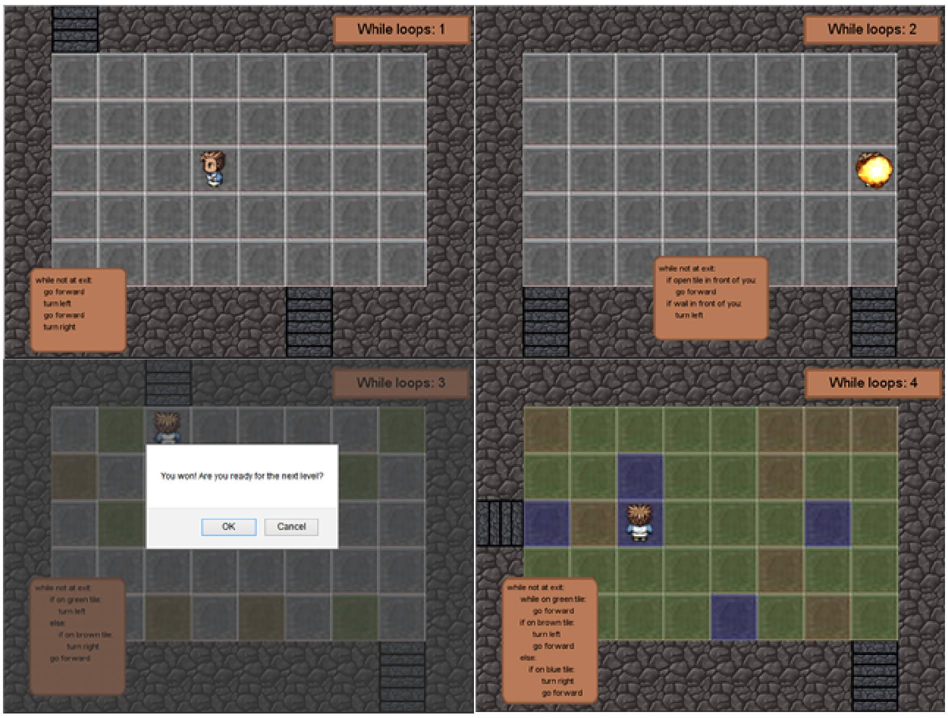

Figure 4.10: The while loop levels

In the final section, algorithmic thinking, the player must write the expressions required to traverse a maze. This requires them to think through the implications of their instructions. The number of instructions is limited to encourage the player to use them as concisely and effectively as possible. The levels also require the players to put repeated instructions into a function, in order to teach pattern detection and another level of abstraction. The instructions for this section were divided into two parts, to allow the player to see the structure of the levels and get used to the controls, before explaining functions. The second level was an abstracted version of the first so that the player could see why the function was useful (requiring seven instructions instead of twelve). The first few levels offer one function for placing repeated instructions in, and the final level offered two functions, that could be called separately.

Figure 4.11: A few of the algorithmic thinking levels

The player enters the instructions using the keyboard or by clicking on the instruction they wish to add. They can undo their last selected move and reset the list of inputted instructions. Once they are happy with their selected instructions, they press run, and the character will carry out the inputted instructions. If those instructions carry the character to the exit, the player moves on. Otherwise they are told to rethink their instructions. To help the player understand the effect their instructions have on the character, the current step executing is highlighted as it is performed (see in Figure 4.11). There is also a step counter for each function block.

Figure 4.12: Emerging from underground

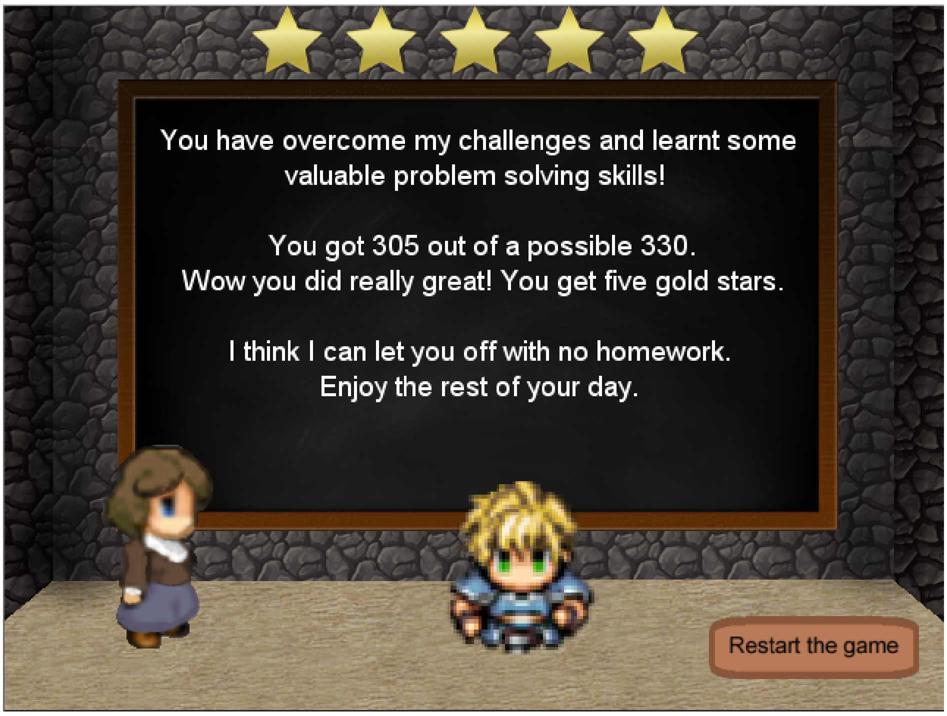

Figure 4.13: The final scene with the player's score

When the player completes this final section, and walks up the ramp of the next piece of corridor on the map, they emerge above ground where the teacher waits in front of the school (Figure 4.12). When they walk up to the teacher, the final scene is shown (Figure 4.13). The player is congratulated on finishing and given their final mark (accompanied by the number of gold stars their mark has earned them).

4.3 Game testing

A pilot study was carried out amongst a variety of peers (n=21),which were made of honours students with a lot of computer science experience who fully understood the concepts and how the game was created (n=8); university students with little to not computer science experience (n=10); and highschool students with no previous experience (n=3). This was in order to test the game as to how it was received and whether there were any unseen errors or bugs. The more experienced testers compiled a Google doc of notes and comments, pointing out spelling errors, bugs and unclear instructions. These notes were taken into account and the game was adjusted accordingly.

4.4 Chapter summary

This chapter describes the structure of the developed game and justifies the technical and structural choices made. Each of the sections in the game (trial and error, if statements, variables, while loops and algorithmic thinking) was explained and screenshots were provided.

The game was tested through a pilot study, where a variety of peers used the game and reported on their experiences.

The next chapter presents the results of the follow up CSc 112 survey, its comparison to the initial survey run as a baseline, and observations made while CSc 112 students played the developed game.

Chapter 5

Intervention and evaluation

To ascertain whether CSc 112 is having an affect on the students' opinions about themselves and about computer science, a second questionnaire was asked near the end of the course to be compared against the initial survey. The survey also asked about the students' experience of playing the game and what their feelings towards it being used as an educational tool.

This chapter also lists observations made while CSc 112 students used the game in a practical. These observations were collected to see how the students interacted with the game, what were common problems that arose, whether there were differences in the way different students played and so on. The scores students achieved while playing were collected and compared by gender.

5.1 Post-intervention survey

This survey was an extended version of the original survey run on the CSc 112 class. It contained all the same questions based on the survey by Wiebe & Miller (2003) in order for them to be directly compared, so any adjustments in the students' attitudes and perceptions of computer science during CSc 112 could be seen. There were additional questions asked that were analyzed by comparing them to the initial survey.

5.1.1 Participants

The survey was run amongst the students enroled in CSc 112 at Rhodes University. 229 students participated, of which almost half were female (50%, n= 115 female, 50%, n= 114 male). Of this group only 19% were considering majoring in computer science (7%, n= 17 females; 12%, n= 27 males). This is consistent with the initial survey run (see Section 3.2.1).

5.1.2 Method and analysis

The game was played by the students in the first CSc 112 programming module practical. The survey was run in October 2014 during the following CSc 112 programming module practical. It asked identical questions to the initial survey (see Section 3.2) as well as additional questions to evaluate the students' experiences of playing the game (a full list of questions can be found in Appendix D[1]). Ethical approval was sought from and approved by the Hamilton Ethics Board (application tracking number: CS14-12). Students were made aware of the intentions and purpose of the study, and that their participation was voluntary and confidential. The survey was conducted electronically using Google Forms and the collected data was stored anonymously in a spreadsheet. The results were statistically analyzed using Statistica. The response scales were converted to ordinal numbers (with strongly agree being given the highest value).

From this data, basic descriptive statistics for the questions related to game use were analysed. A two-way ANOVA with assumed equal variance was performed on the responses of this survey that related to the students' attitudes against the responses in the initial CSc 112 survey to see whether the responses between genders had changed over the course of three months in CSc 112. Three-way crosstabulation frequency analyses were performed on certain questions that indicated significant differences in regard to both time and gender in order to further explore individual questions. Two one-way ANOVAs with assumed equal variance were performed on the questions not previously asked, one testing the null hypothesis that there was no difference between genders and the second testing the null hypothesis that there is no difference between the responses of students choosing to major in computer science, and those choosing not to major.

[1] All responses to the questions asked in the post-intervention survey can be found in electronic format on the accompanying CD-ROM.

5.1.3 Results and discussion

This section provides descriptive statistics of the survey questions related to the game, the one way ANOVAs performed and the two-way ANOVAs performed.

5.1.3.1 Game related questions

Of all the survey participants, 95% played the game (resulting in a new total of n=224: 50%, n=112 female; 50%, n=112 male). Only these students were questioned about the game. Of these participants, 58% greatly enjoyed the game, 19% thought it was ok, 17% enjoyed it but would rather have done something else, and 6% did not enjoy it and found it pointless. Half the participants greatly enjoyed it and only a twentieth did not, this is not conclusive that the game was an enjoyable learning experience, but is a positive indicator that games can be fun.

When asked whether the game explained the concepts clearly, 43% said all concepts made sense, 46% said most concepts made sense, 6% said they understood a few but were mostly guessing and only 3.5% said they made no sense at all. This means 89% of the participating students mostly understood what was being taught by the game, which is a high percentage.

In regard to character choices, 48% of participants said they liked being able to chose their own character and 62% said they chose characters that resembled themselves. This is a slightly lower number than was observed during the practical, but as 19% either did not answer or said they could not remember which character they chose, this could account for the discrepancy. By choosing characters that represent themselves, the students are identifying personally with the game. By personalizing the game, students become more invested in the game which can increase their confidence and determination (Cordova & Lepper, 1996).

The use of games was positively received as an educational tool, with 61% of participants saying the game made learning the concepts really easy and 59% saying more computer science modules should use games for teaching. Only 2% of participants found the game unhelpful and 8% did not think games should be used for teaching computer science.

In response to the question: “The game allowed you to try out the concepts as you were learning them. Do you like learning in context like this, where a concept’s use is explained along side how it works?”, 73% of participants said that this style of learning made more sense to them. 25% were neutral on the subject and 2% did not like this style of learning as it took too much time and they would rather just learn pure concepts. That 2% consists of only 4 students who had all opted to not study computer science.

The game had a positive effect on self confidence for the majority of participants, with 55% of participants stating that the game increased their problem solving confidence and only 5% saying it made their confidence worse. Of the 11 students who said it decreased their confidence, 3 were male and 8 were female. This is a concerning percentage as it implies the game is not completely gender neutral. Alternatively, this discrepancy could be due to females being more easily discouraged due to aforementioned self efficacy issues. However, 5% is still very low and 57% of female participants stated that their confidence improved.

When asked if regular computer science 112 practicals increase their confidence, 46% said yes (46%, n=51 female; 46%, n=51 male). This is lower than the amount that responded well to the game.

The penultimate question was “Did this game increase your interest in Computer Science?”. Only 10.5% said no, with 46% saying yes and 43.5% remaining neutral on the subject. There was a slightly larger number of females than males whose interest was increased (49%, n=55 female; 42%, n=42 male). However, as shown in the next section, this is not a significant difference.

“When you got stuck in the game, how did you handle it?” received 53% of responses that claimed the participants tried a variety of approaches until one worked (58%, n=65 female; 48%, n=54 male). The gender distribution in this question was interesting as this is traditionally a male approach, yet here females make up the majority that used this technique. The answer may lie in the fact that 14% of male participants claimed that they never got stuck in the game. Only 2 females said they did not get stuck in the game. Of the 18% that asked for help, just over half were female, but the distribution was fairly even. Only 19% revisited the instructions when stuck, and this response also had even gender distribution.

In summary, the game was enjoyed by a majority of participants, with 89% of participants understanding most of what was being explained. Two thirds of students liked learning with games and wished more modules used games for teaching. Only 2% found the game unhelpful, a very low percentage. Almost three quarters of the participants enjoyed learning contextually and only 2% did not. The majority of participants felt that the game increased their problem solving confidence (10% higher than the number of participants who felt regular practicals increased their problem solving confidence). Overall the game seems well received and participants were interested in the new learning technique.

5.1.3.2 Investigating responses with regard to gender

A one-way ANOVA was performed on the responses pertaining to the use of the game and new questions not previously asked in the initial survey (see Table 5.1). The questions not previous asked (25 - 30) were omitted as it was felt that the students would have no previous conscious problem solving experience. There was no significant difference between genders for the first two questions. The first asks if they feel what they are learning in the course will be of value, meaning genders are placing equal amounts of importance on the programming. This is a positive result as it means the course itself is being perceived as relevant by both genders. The second question with no significant difference is about continuing to try and solve problems. This result was also confirmed by the question on how students handled getting stuck in the game (discussed above). Significant differences were discovered in questions 27 - 29. These all relate to a drive to solve problems. Males were far more motivated about problem solving than females. This could be a key factor as to why few females enjoy computer science enough to pursue the career: they do not have the same driving need to solve problems.

Table 5.1: One-way ANOVA gender comparison for new questions in the follow up CSc 112 survey

Nr	Question	N	Mean	F	p
25	I will use what I learnt in the programming course in my daily life	215	3.168	2.787	0.096
26	I continue to work at a problem when I can't immediately solve it	216	1.093	1.334	0.249
27	Once I start working on a problem I find it difficult to stop	214	7.638	7.556	**0.006**
28	The challenge of solving problems is something that I find really interesting	212	16.5	17.78	**0.000**
29	I don't understand how people can spend so much time on writing programs and enjoy it	214	8.857	6.737	**0.010**
30	I would rather have someone give me the solution to a difficult problem than work it out myself	216	2.759	2.144	0.145
31	Did you enjoy playing the game?	214	0.171	0.182	0.670
32	Did you feel that the game explained the concepts clearly?	215	0.320	0.748	0.388
33	Did you like being able to choose your own character?	211	1.051	2.902	0.09
34	Did you choose a character that resembled yourself (race or gender)?	174	0.025	0.12	0.73
35	Do you wish some more of the computer science modules used games for teaching?	214	1.019	2.634	0.106
36	The game allowed you to try out the concepts as you were learning them. Do you like learning in context like this, where a concepts use is explained along side how it works?	215	0.783	3.244	0.073
37	Did playing the game help increase your confidence in being able to problem solve?	214	0.005	0.013	0.909
39	Do the normal practicals for Computer Science 112 increase your self confidence in being able to problem solve?	213	0.041	0.111	0.740
40	Did this game increase your interest in Computer Science?	213	0.734	1.662	0.199
41	When playing, did you rely more on reading the instructions or figuring out how to do things on your own?	213	4.243	7.181	**0.008**

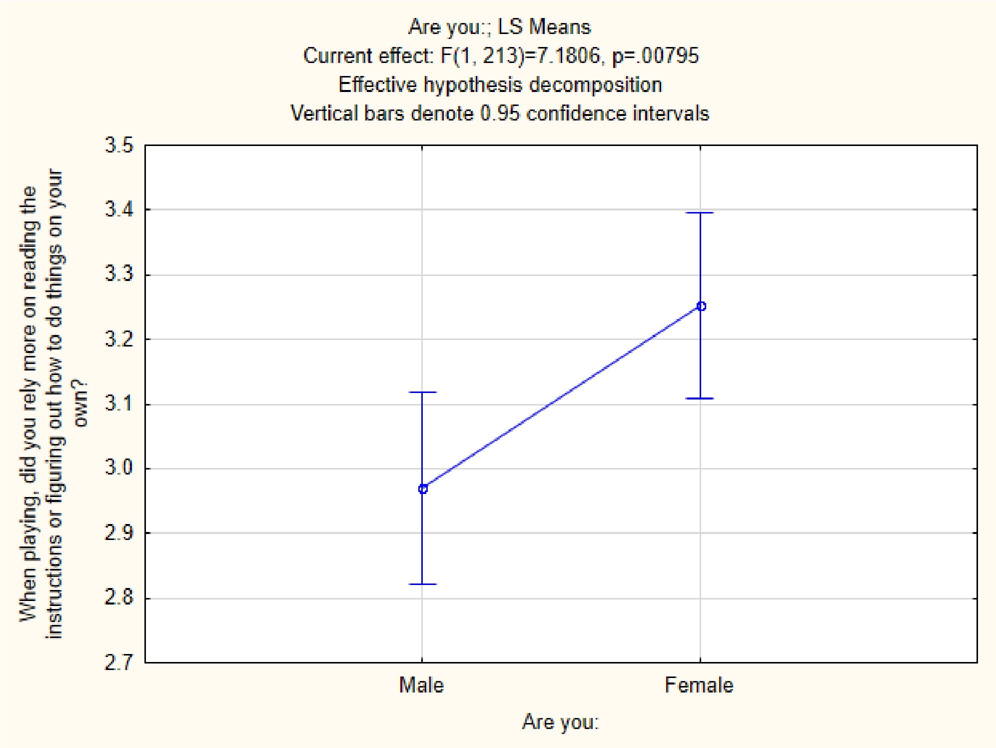

Figure 5.1: One-way ANOVA of question 46: When playing, did you rely more on reading the instructions or figuring out how to do things on your own?

The only question that showed a significant difference between genders in regard to the game was 41: "When playing, did you rely more on reading the instructions or figuring out how to do things on your own?". The significance of this is that it confirms the findings of Jorgensen *et al.* (2013) that females rely more heavily on instructions than males (as seen in Figure 5.1). The fact that there are no other significant differences in responses means that the null hypothesis is true in all other questions. This indicates that the game affected both genders equally, with both genders responding in the same way to it.

5.1.3.3 Investigating majoring students vs non-majoring students

Table 5.2: One-way ANOVA to compare choice to major in computer science on game related questions

Nr	Question	N	Mean	F	p
31	Did you enjoy playing the game?	220	0.012	0.013	0.911
32	Did you feel that the game explained the concepts clearly?	221	0.893	2.117	0.147
33	Did you like being able to choose your own character?	217	0.148	0.407	0.524
34	Did you choose a character that resembled yourself (race or gender)?	180	0.076	0.366	0.546
35	Do you wish some more of the computer science modules used games for teaching?	220	0.172	0.425	0.515
36	The game allowed you to try out the concepts as you were learning them. Do you like learning in context like this, where a concepts use is explained along side how it works?	221	0.035	0.142	0.707
37	Did playing the game help increase your confidence in being able to problem solve?	219	0.681	1.949	0.164
38	Do the normal practicals for Computer Science 112 increase your self confidence in being able to problem solve?	219	0.42	1.138	0.287
39	Did this game increase your interest in Computer Science?	219	0.146	0.332	0.565
40	When playing, did you rely more on reading the instructions or figuring out how to do things on your own?	219	0.782	1.277	0.260

The response to the first question (rate the statement "I plan to major in computer science" based on how you feel) was grouped into two categories for this test. 'Strongly agree' and 'Agree, but with reservations' were marked as 'Yes' and the rest as 'No'. The results are in Table 5.2. There were no significant differences between the two groups, with all p-values >0.05. In all observed cases, the variation of answers amongst students considering majoring was far larger than for students choosing not to major. This indicates that the use of the game was not affected by the choice to major in computer science.

5.1.3.4 Comparing the first and second survey run on CSc 112

Table 5.3 is a summary of a two-way ANOVA done on each question that overlapped between the two surveys run within CSc 112. It shows the effect a term's worth of CSc 112 has had on the students' attitudes towards computer science. It must be noted that because the number of participants in this survey is less than the first survey, mean results could be skewed by the difference in actual participants. However, the ratio of genders is the same between the surveys is the same. Also, the percentage of students majoring in computer science amongst the participants is very similar to the first survey, so the results should be representative of the whole class.

Of all the questions, only one saw a significant p-value of the interaction between gender and time. This was question 9 ("I enjoy solving problems"). In this question, male and female students' responses changed inversely, with female students lowering their answer (from a mean of 3.65 to 3.32) and male students increasing theirs (from a mean of 3.9 to 3.97). This effect can be seen more clearly in Figure A.9. This means that CSc 112 is causing females to lose an enjoyment for problem solving, when it should be doing the inverse. It is indeed having the inverse effect on males. This acts as confirmation of the idea that the way computer science is being taught appeals more to male students.

Of all the questions asked, fifteen of the twenty-four questions had a significant p-value (<0.05) when comparing the answers of the two genders. Nine questions had a significant p-value for comparing the answers between the two surveys (there had been a change over time). Only 3 had significant p-values for both comparisons (questions 9, 23, 24). Question 9 has already been discussed, and so the other two questions with both p-values will be discussed.

Question 23 ("I try and avoid computers as much as I can") sees females agreeing more strongly with this statement than males, though the female responses still having a mean that falls within "Disagree, but with reservations". Between the first and second survey, there has been a significant increase in the number of students agreeing with this statement. A three-way crosstabulation frequency analysis[2] (shown in Table 5.4) revealed the changes in the number of students answering each question. In the first survey, 58% of females selected strongly disagree, in the second only 44% selected this option. Males went from 69.5% strongly disagreeing to 59%. The number of females who agree that they avoid computers almost doubled between surveys and the number of males who were

[2]In this table, the column percent indicates the percentage that that record makes up of a specific answer (e.g. females from the first survey that answered "Strong disagree" out of all the students who answered "Strongly disagree") and row percent indicates the percentage of that record out of that row (eg. percentage of all females from the first survey that answered "Strongly disagree").

Table 5.3: Comparative two-way ANOVA between the two CSc 112 surveys. Univariate Tests of Significance

Nr		MS	F	p
1	Timestamp	0.053	0.031	0.860677
	Gender	31.828	18.534	**0.000019**
	Timestamp*Gender	1.287	0.749	0.387063
2	Timestamp	0.003	0.002	0.962764
	Gender	42.033	33.604	**0.000000**
	Timestamp*Gender	2.100	1.679	0.195537
3	Timestamp	0.036	0.034	0.852847
	Gender	51.883	50.218	**0.000000**
	Timestamp*Gender	1.283	1.241	0.265613
4	Timestamp	2.517	1.489	0.222870
	Gender	53.915	31.892	**0.000000**
	Timestamp*Gender	0.006	0.003	0.953203
5	Timestamp	0.021	0.015	0.901944
	Gender	41.057	30.121	**0.000000**
	Timestamp*Gender	0.797	0.585	0.444700
6	Timestamp	0.119	0.094	0.758796
	Gender	7.214	5.723	**0.017041**
	Timestamp*Gender	4.488	3.560	0.059655
7	Timestamp	2.54	3.52	0.061109
	Gender	1.83	2.54	0.111664
	Timestamp*Gender	1.96	2.72	0.099702
8	Timestamp	8.311	8.339	**0.004013**
	Gender	5.273	5.291	**0.021768**
	Timestamp*Gender	1.342	1.347	0.246330
9	Timestamp	29.458	32.690	**0.000000**
	Gender	2.525	2.802	0.094621
	Timestamp*Gender	5.744	6.374	**0.011824**
10	Timestamp	3.05	4.17	**0.041472**
	Gender	0.97	1.33	0.249419
	Timestamp*Gender	0.40	0.55	0.459524
11	Timestamp	0.656	0.438	0.508505
	Gender	1.527	1.019	0.313252
	Timestamp*Gender	0.210	0.140	0.708039
12	Timestamp	0.464	0.285	0.593388
	Gender	0.053	0.032	0.857347
	Timestamp*Gender	0.683	0.419	0.517454
13	Timestamp	0.09	0.081	0.776427
	Gender	10.08	8.597	**0.003489**
	Timestamp*Gender	0.40	0.341	0.559500
14	Timestamp	0.71	0.66	0.415288

	Gender	4.90	4.62	**0.032044**
	Timestamp*Gender	1.59	1.50	0.221660
15	Timestamp	4.914	3.807	0.051493
	Gender	7.284	5.642	**0.017831**
	Timestamp*Gender	0.628	0.487	0.485635
16	Timestamp	4.607	2.964	0.085651
	Gender	6.484	4.171	**0.041526**
	Timestamp*Gender	1.217	0.783	0.376554
17	Timestamp	5.064	4.826	**0.028402**
	Gender	3.610	3.440	0.064105
	Timestamp*Gender	1.528	1.456	0.227966
18	Timestamp	6.12	10.56	**0.001220**
	Gender	0.00	0.00	0.978629
	Timestamp*Gender	0.02	0.03	0.868369
19	Timestamp	8.897	10.47	**0.001279**
	Gender	0.117	0.14	0.710208
	Timestamp*Gender	0.017	0.02	0.887035
20	Timestamp	0.680	0.508	0.476130
	Gender	2.021	1.512	0.219316
	Timestamp*Gender	1.773	1.327	0.249848
21	Timestamp	23.885	25.256	**0.000001**
	Gender	1.111	1.175	0.278875
	Timestamp*Gender	0.098	0.104	0.747145
22	Timestamp	0.017	0.015	0.903143
	Gender	4.433	3.883	**0.049216**
	Timestamp*Gender	2.139	1.874	0.171516
23	Timestamp	6.853	5.965	**0.014866**
	Gender	8.723	7.593	**0.006031**
	Timestamp*Gender	0.208	0.181	0.670917
24	Timestamp	6.569	6.946	**0.008611**
	Gender	6.439	6.809	**0.009288**
	Timestamp*Gender	0.101	0.107	0.743596

neutral to the statement almost tripled from 6.5% to 19% between surveys. This means that CSc 112 is resulting in students becoming more opposed to using computers, instead of making them more comfortable so that they can utilize them more.

A similar result can be seen in question 24 ("I am excited to learn more about computers"). The graph seen in Figure A.24 is the inverse of question 23 as it is a positively phrased question instead of a negatively phrased one. Both genders have lowered their amount of excitement to learn about computers. Males are reduced to the level of the females in the first survey and females have moved closer to being ambivalent and neutral about the

Table 5.4: Three-way crosstabulation frequency table of question 23: I try and avoid computers as much as I can

	Time	Gender	Strongly disagree	Disagree, but with reservations	Neutral	Agree, but with reservations	Strongly agree	Row totals
Count	First	Female	118	40	27	8	9	202
Column Percent			45.91%	55.56%	67.50%	42.11%	64.29%	50.25%
Row Percent			58.42%	19.80%	13.37%	3.96%	4.46%	63.92%
Count	First	Male	139	32	13	11	5	200
Column Percent			54.09%	44.44%	32.50%	57.89%	35.71%	49.75%
Row Percent			69.50%	16.00%	6.50%	5.50%	2.50%	63.90%
Count	Total		257	72	40	19	14	402
Column Percent			68.72%	59.50%	49.38%	61.29%	63.64%	
Row Percent			63.93%	17.91%	9.95%	4.73%	3.48%	
Count	Second	Female	50	31	20	7	6	114
Column Percent			42.74%	63.27%	48.78%	58.33%	75.00%	50.22%
Row Percent			43.86%	27.19%	17.54%	6.14%	5.26%	36.08%
Count	Second	Male	67	18	21	5	2	113
Column Percent			57.26%	36.73%	51.22%	41.67%	25.00%	49.78%
Row Percent			59.29%	15.93%	18.58%	4.42%	1.77%	36.10%
Count	Total		117	49	41	12	8	227
Column Percent			31.28%	40.50%	50.62%	38.71%	36.36%	
Row Percent			51.54%	21.59%	18.06%	5.29%	3.52%	
Count	Column Total		374	121	81	31	22	629

statement. Computer science should act to excite students. Though CSc 112 is not a true and pure reflection of computer science, this is still a negative reflection on the course.

Some of the 15 questions that saw a significant gender difference can be grouped into a topic that genders disagree on. The first of these groupings is general statements about programming and computer experience (questions 1-6, 8). Questions 1 (Figure A.1), 2 (Figure A.2), 3 (Figure A.3), 5 (Figure A.5) and 6 (Figure A.6) show males increasing their rating of the questions and females decreasing their ratings. This inverse effect seems to show that females are losing confidence in themselves and the subject, while male students do the opposite. The difference over time did not produce a significant p-value, but the trend is still present. Question 4, “I already have programming experience” (Figure A.4), was one of the few questions to see almost parallel growth, which is good as it means both genders feel they have acquired programming knowledge, even if males claim to have more experience than females. In question 8, both genders agreed with the statement ‘When I find a problem difficult, I usually just give up’ (Figure A.8) in the second survey. In the first survey, the question had no significant difference between the responses of both genders. This changed in the second survey, with females becoming increasingly more likely to agree than males. CSc 112 has caused the students to lose faith in their ability to eventually solve a problem if they keep trying.

The second group of questions to disprove the null hypothesis that there is no difference between the responses of the two genders were the questions concerning the gender divide in computer science (questions 13 - 16). In question 13 (Figure A.13), the statement “Females are as good as males at programming” gained almost the same mean from females but males raised their response by almost 0.1 which shrunk the difference between gender responses. However, the next question “Studying programming is just as appropriate for women as for men” has females lowering the mean response by almost 0.2 and males remaining almost the same. This is due to the increased number of females that selected “Agree, but with reservations” (11% of females increased to 18%) and a decrease in the number that selected “Strongly agree” (78% of females down to 67%). Females remaining neutral to the statement also increased (5% to 11%). This means that though females are increasingly seen as capable of studying programming (question 13), females are finding it less appropriate for them to study once they participated in the CSc 112 course. Both question 15 (“It is hard to believe a female would do as well as males in programming”) and 16 (“It makes sense that there are more men than women in computer science”) initially had significant p-values in the first survey but this difference in opinion decreased in the second survey. Both statements were agreed with more by both genders (there was a larger change in female responses). These two statements pose questions that are opposed to females being capable of doing computer science and programming. CSc 112 is decreasing both genders' opinions of females in computer science.

In question 22 ("Programming is boring" as seen in Figure A.22) both genders initially agreed on a response but in the second survey females found programming more boring and males found it less boring. The genders responded in an almost exactly opposite way. This could indicate that the way programming is being portrayed is currently only appealing to males, providing confirmation of the idea that computer science is presented in a gender biased way.

Question 17, which states that programming is an important skill (Figure A.17), only has a significant difference over time, not gender. Originally the genders had almost the same mean response value but in the second survey, the mean female response decreased more than the male, suggesting that time had a strong effect on the student responses. The course is decreasing the perceived importance of programming amongst the students. Interestingly, question 18, "Problem solving is a very important skill to have" has no significant difference in regard to gender. The mean response for both genders is almost identical. Problem solving is seen as important, but both genders thought less of the skill after three months of CSc 112.

Though question 7 ("I think I can learn programming") has no significant p-values for time or gender, it is a question that is worth discussing. Female students' responses decreased from a mean response of 4.2 to 3.96. Though this is a small movement that is not considered significant, when compared to the male shift of 0.1, it must be acknowledged that females have lost confidence in themselves, even if only slightly. In the first survey, not a single female strongly disagreed that they could learn programming, and 1.5% of males did. In the second survey, 3% of females selected strongly disagree and only 1% of males did. These small change in the male percentages could, however, have been caused by the differing number of participants. In the first survey 44% of females strongly agreed that they could learn programming but after three months of the course, only 27% of females felt the same way. The number of male students to strongly agree also decreased, from 45% to 39%.

The only other two questions to have no significant p-values for time or gender were 11 (Figure A.11) and 12 (Figure A.12). Both of these refer to how students deal with success in the subject. Both questions were negatively phrased and had a mean answer of "Disagree, but with reservations". This is encouraging as both genders do not see computer science achievement as a negative thing.

In conclusion, there was a correlation between CSc 112 and negatively affected student attitudes. Students were less interested in the subject and had less confidence in themselves. That CSc 112 is the cause of this attitude change cannot be proven, as there are too many other factors (such as other subjects or students becoming more stressed near the end of term) to isolate CSc 112 as the cause. However, it is a strong correlation. This trend is particularly obvious amongst female students, whose attitudes got worse fairly consistently through every question. This shows that females are indeed being demotivated, if not by the current computer science course than by some other university factor. It must be acknowledged that CSc 112 is an introductory course and contains a lot of modules that do not pertain to computer science directly, more to information technology, but it is still an important discovery. CSc 112 is driving students away from the computer science department.

5.1.4 Summary

The game was enjoyed by the majority of participants, with almost all participants stating they understood almost all of what was being explained in the game. A large majority liked learning with games and wished more modules in computer science used this approach. Three quarters of students enjoyed learning contextually and the majority of participants felt it increased their problem solving confidence.

Comparing the responses of the two genders for new attitude questions revealed a large difference in how motivated each gender was to solve problems and how much drive they have for working on problems. That females showed less enthusiasm for enjoying problem challenges could offer insight into why fewer females do not remain in the field.

Comparing the responses of the two genders for the game related question revealed only one question had a significant difference. This difference was what students relied on more when trying to solve the problems: instructions or intuition. This difference confirms what was found by Jorgensen *et al.* (2013). That the rest of the questions showed no significant difference means the game appealed to both genders in the same way, and both genders responded to it in the same way. This shows that the use of games is a gender neutral teaching technique. This was further confirmed when there was no significant difference between the scores achieved by each gender.

When comparing majoring students again non-majoring students for questions related to the game, no significant difference was found. This implies that the game had a positive effect on all skill levels.

The comparison using a two-way ANOVA revealed that there was a strong correlation between negative attitudes and CSc 112 though it cannot be isolated as the cause to due the number of exterior attitudes that could also affect students. The decrease in confidence and interest was more prevalent amongst females.

5.2 Observations made during game use

The game was used by the CSc 112 students during the introductory Programming Logic practical from the 29 September to 3 October 2014. During this time the students were observed in order to see how they played and interacted with the game. This section will present a summary of the notes made during three of the five practicals.

- Students mostly chose the character most like themselves (with both gender and race); very few exceptions were noticed.

This echoes results found by the follow up survey and indicates an element of self identification; the students are placing themselves within the game.

- Students were often very quick to ask their tutors or friends for help, instead of trying to solve the problem themselves or going back to the instructions.
- More students than had originally been assumed were reading the instructions for each section, of both genders.
- Dialog boxes were not acknowledged, especially if they contained instructions (as they did for two levels). However, the instruction rooms were read far more thoroughly.

This last comment was noted on a large scale. Future works should take care to make instructions obvious, so that even they are skipped over, it is a conscious choice.

- Female students appeared to be more frustrated by negative feedback but no cases of self blame were noticed.
- Female students were noticed to be more likely to collaborate and work together, but a few male students did work with friends.

- A few students remarked when asked that they would be unable to play the game without the instructions and they liked being able to immediately test out what they had learnt.
- Many students found the very first level (Figure 4.5) frustrating and were very quick to give up when they received no immediate feedback. This could be due to the steep escalation in game difficulty and perhaps the escape games should have been placed first. The lack of instructions made students feel abandoned.

The last two points emphasis the importance of written instructions. Jorgensen *et al.* (2013) noted that females were far more heavily reliant on instruction, but it was found that even if this was true, male students still wanted initial guidance, even if they are less likely to ask for help.

- The if statement level that had changing if statements for each item had its instructions in a dialog box, and so as a result many did not start rereading the if statements for each item until they received negative feedback.
- The variable level that required the statements to be put in order also had its instructions in a dialog box. It was widely misunderstood and often students asked for help. In future, this level should be reworked.
- Many students became quickly frustrated at having to do mental arithmetic in the variables levels.
- Even though the difference between a mathematical equals sign and the assignment equals sign was explicitly stated in the instructions, the two were still confused. Perhaps a more interactive approach needs to be taken for this concept.
- In the algorithmic thinking section, some students carefully worked out the steps before attempting the level, while others used pure trial and error. This was not gender specific.
- The most questions were asked in the variable section, even with the instructions being divided up. Students really struggled to identify what could be repeated. A possible solution to this could have been including a stepping control, where students could step through the instructions individually to track them more precisely.
- Most students took between 30 and 40 minutes to complete the game. Two students took two hours. As this information was reported by a tutor, these students were not individually spoken to as to why it took them so long.

- Upon completing the game two noticeable reactions were observed: either excitement at having finished or disappointment that they had to return to regular practical work.
- Comments similar to "the game was annoying but fun" or "difficult or enjoyable" were offered up by students.
- A few students commented that they preferred the game to regular practicals, and some replayed to improve their score (and to procrastinate).
- There appeared to be more questions from minorities (students of colour and female students). This could possibly be related to confidence and experience levels.
- Students were more likely to turn to their peers for help than their tutors.
- When a students failed a level the reaction usually consisted of looking at their neighbour's game, thinking about the problem, possibly trying again, and then asking for help. Perhaps more hints or assistance should be made available so the game can exist in isolation, without tutors they can turn to for help.

Future works could create multiplayer interactive games to encourage this behaviour that occurs naturally, especially as the game should be able to exist outside of a tutor assisted practical.

- The tutors controlling the practicals were excited by the game, and a few of them played it, despite it being far below their skill level. However, students who had previous experience were frustrated by having to play a game that was below their skill level. There is scope for a higher ceiling.

Ultimately the game was well received and almost all the students were enthusiastic about playing the game. The concepts were understood and nearly all students finished the game.

5.2.1 Analyzing the scores achieved in the game

The tutors that took the practicals were asked to write down the score students achieved when they played the game. This was done to explore whether there was a difference between genders in how well they did in the game. A one-way ANOVA was performed on the collected marks. Though females had a slightly lower mean score (297 against a male mean of 302), no significant difference was found between genders. This again emphasizes that the game was gender neutral.

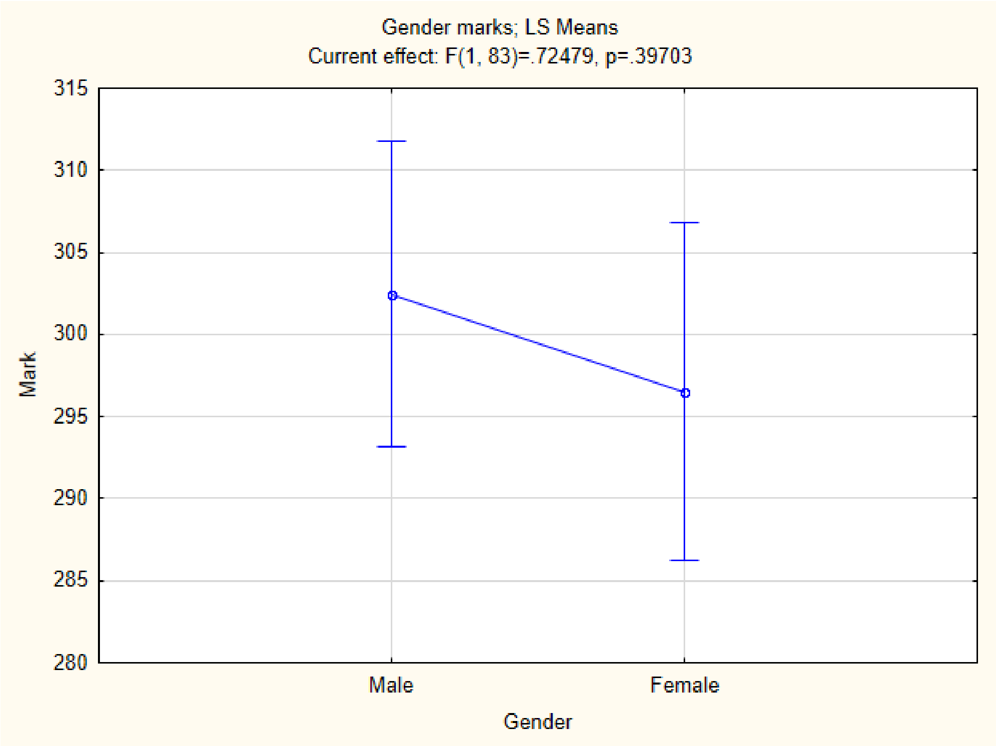

Figure 5.2: One-way ANOVA of scores achieved in game play

5.3 Chapter summary

This chapter outlines the design, responses and analysis of the post-intervention survey run on CSc 112 students as well as observations made while students played the game.

The game was enjoyed by the majority of students and taught the necessary concepts to a large majority of students. Participants rated the effect the game had had on their self confidence as higher than regular CSc 112 practicals. The game was shown to be gender neutral and the responses to it were not affected by previous experience in the field.

Female mean responses rated the participants' self efficacy and interest levels lower in the second survey than in the initial survey. Male mean responses increased in many of the same questions. Overall, no direct link between the change in attitudes within the students and CSc 112 could be found.

Observing the CSc 112 students play the game suggested areas for future work (such as multiplayer versions of the game). It also showed the game was well received and that students were internalizing the game through personalization.

The next and final chapter presents the conclusions drawn from this research as well as suggesting future work within the field.

Chapter 6

The final verdict

6.1 Conclusion

Computer science university courses are definitely seeing a decline in enrollment rates, with the ratio of male to female students at a staggeringly low number. The amount of female and male participants from the interdepartmental survey confirm these low numbers (with only 10% of participants being females studying computer science). The initial survey revealed that computer science is not viewed as a fun and interesting subject, and so has a poor reputation amongst students. Due to the positive responses to playing the developed game, games could help improve the popularity of computer science, which would encourage more students to enrol. The results of the comparison of the two attitude surveys showed that the introductory computer science course at Rhodes University is acting to discourage students from the field, particularly females who rate their own confidence levels as lower after the duration of the course. The attitude surveys also indicate that stereotypical views of females in computer science are common, and CSc 112 is not acting against these biases. With a large majority of questions within the attitude section of the surveys showing significant differences between genders, it must be concluded that males and females are viewing and interpreting the subject differently.

Prior research has shown that contextualized learning provides a gender neutral approach to teaching computer science in a way that could increase the subject's perceived importance and reputation. The use of games was positively received and both genders gave similar feedback, indicating that the use of games to teach concepts is a gender neutral technique. The majority of questioned students enjoyed the game and wished more courses utilized this mode of teaching. To build a comprehensive and instructive game that has a complex and engaging narrative that could fully utilize the benefits of these findings would take further time and resources. However, even the relatively simple game built for this project was well received.

6.2 Achievement of research goals

The main goal of this research was to develop a game (see Section 4.2) to teach computational thinking skills in contextualized way, and to assess whether this would have a positive affect on how students perceive their own abilities and computer science in general. The game was developed in accordance with the needed criteria and was used by students enroled in CSc 112. The results of how the game was received (as explored in Section 5.1.3.1) indicated that the game was well received and enjoyed. Most students liked contextualized learning and the use of games for learning. The majority of students felt the game increased their confidence to solve problems. This indicates that games make an effective teaching tool and the contextualized learning is greatly appreciated by the students.

The second research goal was to explore how females responded to contextualized learning through the use of games, to test if this technique could be used to combat the gender divide inherent in computer science. Only one of all questions asked about the game had a significant difference between the responses of each gender (see Section 5.1.3.2). This means that both males and females are responding in the same way to the game and that this is a potentially gender neutral teaching technique.

The research aimed to assess how the current CSc 112 curriculum was affecting the attitudes of students. Though direct causation could not be proven, there was a provable decline in some areas, more so for females than males (as explored in Section 5.1.3.4). It is unclear which area of the students' lives could be causing this, but as less than 50% of participants in the second survey said that normal computer science practicals increased their problem solving confidence, some of this decline should be attributed to CSc 112.

A subgoal of this research was to determine what has been done in previous research in regard to contextualized learning, especially with the focus on its effect on gender and being implemented through the use of games. This was thoroughly explored in Chapter 2. Many successful examples of contextualized learning (see Section 2.2) and using games to teach computer science (see Section 2.3) were found. The combination of these techniques create effective teaching tools that can dynamically engage and entertain the student while they learn. It also was shown to be a gender neutral technique that was well received by both genders (see Section 2.4.1).

The current perceptions of contextualized learning and the lack of gender diversity in the field within the Rhodes University computer science department were ascertained in Section 3.1.3. The general findings were that computer science at Rhodes University is slightly biased to male learning (from a mostly female perspective). As for contextualized learning, there were contradictory opinions, but generally most were open to the inclusion of the technique into the framework of regular teaching practices.

All research goals were achieved with varying degrees of success. The main objective, to develop a game to teach computational thinking contextually and to assess its success, was achieved within the scope of the research. More research into all of the above-mentioned goals would be beneficial to computer science tertiary education.

6.3 Future works

This project has a lot of scope for future work. As mentioned in the above section, more complex games could be developed that would teach concepts more thoroughly and perhaps include a test framework to allow the progress of individual students to be tracked. The effect of more complex and engaging narratives should also be explored. A multi-player game could also be developed as many students naturally worked together when playing the game.

Comments from the interdepartmental survey indicated a lack of core skills from students leaving high school. Future works should explore how these skills, especially universal computational thinking skills, could be integrated into highschool, or even primary school, curriculums. By addressing the problem in highschool, it may be easier to adjust gender perceptions and stereotypes that have already become ingrained at university level.

Future works should consider interviewing students (particularly female students) who chose to leave the field. They could provide insights into the short falls of computer science tertiary education and reveal what about computer science is discouraging (female) students.

References

Anon. 2009. Gender Differences in Science Achievement. Online. Available from: https://www.engr.psu.edu/AWE/misc/ARPs/ARP_InfoSheet_Science.pdf. <Accessed 26/02/2014>.

Bandalos, D. L., Yates, K., & Thorndike-Christ, T. 1995. Effects of math self-concept, perceived self-efficacy, and attributions for failure and success on test anxiety. *Journal of Educational Psychology*, **87**(4), 611.

Bayliss, Jessica D. 2009. Using games in introductory courses: tips from the trenches. *Pages 337–341 of: ACM SIGCSE Bulletin*, vol. 41. ACM.

Beyer, S., Rynes, K., Perrault, J., Hay, K., & Haller, S. 2003. Gender differences in computer science students. *Pages 49–53 of: ACM SIGCSE Bulletin*, vol. 35. ACM.

Bundy, A. 2007. Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, **1**(2), 67–69.

Burge, J.E., Gannod, G.C., Doyle, M., & Davis, K.C. 2013. Girls on the Go: A CS Summer Camp to Attract and Inspire Female High School Students. *Pages 615–620 of: SIGCSE 2013: The changing face of computing.*

Carbonaro, Mike, Szafron, Duane, Cutumisu, Maria, & Schaeffer, Jonathan. 2010. Computer-game construction: A gender-neutral attractor to Computing Science. *Computers & Education*, **55**(3), 1098–1111.

Carter, L. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science. *ACM SIGCSE Bulletin*, **38**(1), 27–31.

Casey, B., Erkut, S., Ceder, I., & Young, J.M. 2008. Use of a storytelling context to improve girls' and boys' geometry skills in kindergarten. *Journal of Applied Developmental Psychology*, **29**, 29–48.

Casey, M. B., Nuttall, R. L., & Pezaris, E. 1997. Mediators of gender differences in mathematics college entrance test scores: a comparison of spatial skills with internalized beliefs and anxieties. *Developmental psychology*, **33**(4), 669.

Chamillard, A. T., & Karolick, D. 1999. Using learning style data in an introductory computer science course. *ACM SIGCSE Bulletin*, **31**(1), 291–295.

Chang, J. K., Dang, L. H., Pavleas, J., McCarthy, J. F., Sung, K., & Bay, J. 2012. Experience with Dream Coders: developing a 2D RPG for teaching introductory programming concepts. *Journal of Computing Sciences in Colleges*, **28**(1), 227–236.

Clewell, B., & Campbell, P. B. 2002. Taking stock: Where we've been, where we are, where we're going. *Journal of Women and Minorities in Science and Engineering*, **8**, 255–284.

Cordova, D. I., & Lepper, M. R. 1996. Intrinsic Motivation and the Process of Learning: Beneficial Effects of Contextualization, Personalization, and Choice. *Journal of Educational Psychology*, **88**(4), 715–730.

Dickey, M. D. 2006. Game design narrative for learning: Appropriating adventure game design narrative devices and techniques for the design of interactive learning environments. *Educational Technology Research and Development*, **54**(3), 245–263.

Geddis, A.N. 1991. Improving the quality of science classroom discourse on controversial issues. *Science Education*, **75**(2), 169–183.

Gee, J. P. 2003. What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, **1**(1), 20–20.

Gunn, C., French, S., McLeod, H., McSporran, M., & Conole, G. 2002. Gender issues in computer-supported learning. *Journal of Association for Learning Technology*, **10**(1), 32–44.

Henin, Steven. 2014. When Women Stopped Coding. Online. Available from: `http://www.npr.org/blogs/money/2014/10/21/357629765/when-women-stopped-coding?utm_campaign=storyshare&utm_source=twitter.com&utm_medium=social`. <Accessed: 23/10/2014>.

Holman, J., & Pilling, G. 2004. Thermodynamics in Context: A Case Study of Contextualized Teaching for Undergraduates. *Journal of Chemical Education*, **81**(3), 373–375.

Horne, J. 2007. Gender Differences in computerised and conventional educational tests. *Journal of Computer Assisted Learning*, **23**, 47–55.

Jorgensen, R., Logan, T., & Lowrie, T. 2013. Navigating and decoding dynamic maps: Gender preferences and engagement Differences within- and outside-of game experiences. *Australian Journal of Educational Technology*, **29**(5), 626–637.

Koul, R., & Dana, T.M. 1997. Contextualized Science for Teaching Science and Technology. *Interchange*, **28**(2-3), 121–144.

Lau, W.W.F., & Yuen, A.H.K. 2010. Gender differences in learning styles: Nurturing a gender and style sensitive computer science classroom. *Australasian Journal of Educational Technology*, **26**(7), 1090–1103.

Laurel, B. 1998. *From Barbie to Mortal Kombat: Gender and computer games.* MIT Press.

Leutenegger, S., & Edgington, J. 2007 (March). A game first approach to teaching Introductory programming. *Pages 115–118 of: Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education.*

Lu, J. J., & Fletcher, G. H. L. 2009 (March). Thinking about Computational Thinking. *Pages 260–264 of: Proceedings of the 40th SIGCSE Technical Symposium on Computer Science Education.*

Malone, Thomas W. 1981. Toward a theory of intrinsically motivating instruction. *Cognitive science*, **5**(4), 333–369.

Margolis, J., & Fisher, A. 2002. *Unlocking the clubhouse: Women in Computing.* The MIT Press.

Mishra, A. 2003. Age and school related differences in recall of verbal items in a story context. *Social Science International*, **19**, 12–18.

Mitchell, R. L. 2013. Women computer science grads: The bump before the decline. Online. Available from: `http://www.computerworld.com/article/2474991/it-careers/women-computer-science-grads--the-bump-before-the-decline.html`. <Accessed 23/10/2014>.

Muratet, M., Torguet, P., Jessel, J., & Viallet, F. 2009. Towards a serious game to help students learn computer programming. *International Journal of Computer Games Technology*, **2009**, 3.

Oers, B. Van. 1998. From Context to Contextualizing. *Learning and Instruction*, **8**(6), 473–488.

Parker, L.E., & Lepper, M. R. 1992. Effects of Fantasy Contexts on Children's Learning and Motivation: Making Learning More Fun. *Journal of Personality and Social Psychology*, **62**(4), 625–633.

Prensky, M. 2002. The motivation of gameplay: The real twenty-first century learning revolution. *On the Horizon*, **10**(1), 5–11.

Repenning, A., Webb, D., & Ioannidou, A. 2010. Scalable game design and the development of a checklist for getting computational thinking into public schools. *Pages 265–269 of: Proceedings of the 41st ACM technical symposium on Computer science education*. ACM.

Robinson, John A, & Hawpe, Linda. 1986. *Narrative thinking as a heuristic process.* Praeger Publishers/Greenwood Publishing Group.

Rosser, S. 1990. *Female-friendly science: Applying women's studies methodds and theories to attract students.* Pergamon Press.

Shaw, G., & Marlow, N. 1999. The role of student learning styles, gender, attitudes and perceptions on information and communication technology assisted learning. *Computers & Education*, **33**, 223–234.

Tartre, L. A. 1990. Spatial skills, gender, and mathematics. *Mathematics and gender*, 27–59.

Wiebe, E. N., Williams L. Yang K., & Miller, C. 2003. *Computer science attitude survey.* North Carolina State University: Raleigh.

Wing, J. M. 2008. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society*, **366**(July), 3717–3725.

Yager, R.E. 1996. *Science/technology/society as reform in science education.* State University of New York Press.

Zweben, Stuart. 2012. Computing Degree and Enrollment Trends. Online. Available from: `http://cra.org/uploads/documents/resources/taulbee/CRA_Taulbee_CS_Degrees_and_Enrollment_2011-12.pdf`. <Accessed: 23/10/2014>.

Appendix A

Two-way ANOVA results for surveys run on CSc 112

These two-way ANOVAs compare the effects of time and gender on the students' responses to the attitudes surveys at the beginning of the CSc 112 course and the end of the programming course. Bear in mind that the y-axis scale changes between graphs but the grid lines remain in proportion.

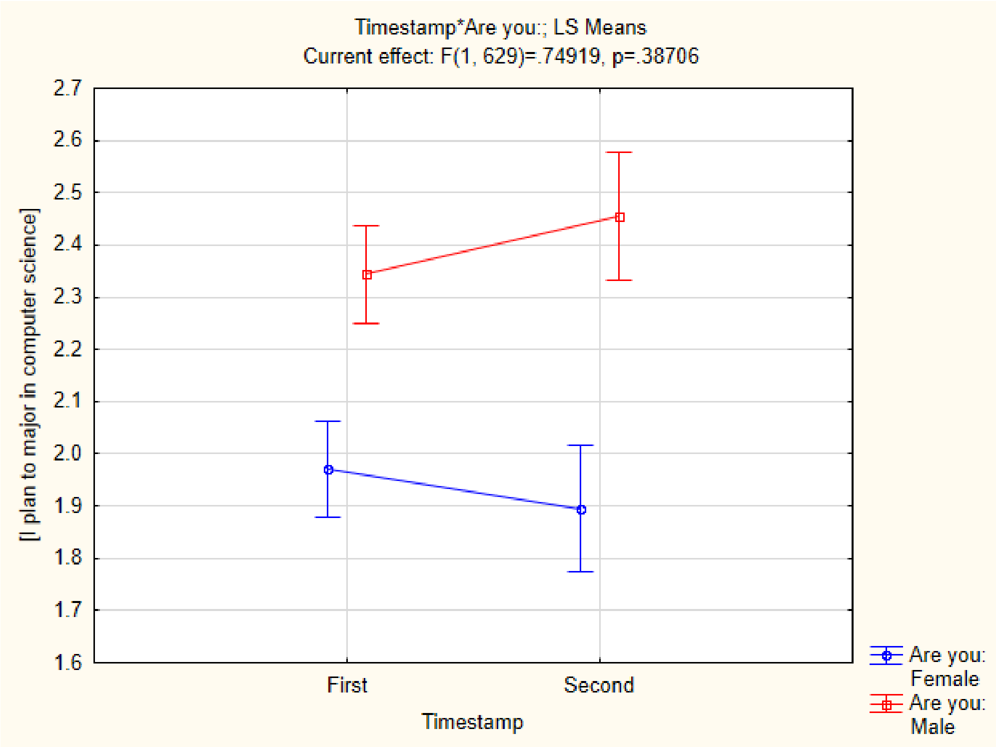

Figure A.1: Two-way ANOVA results for: I plan to major in computer science

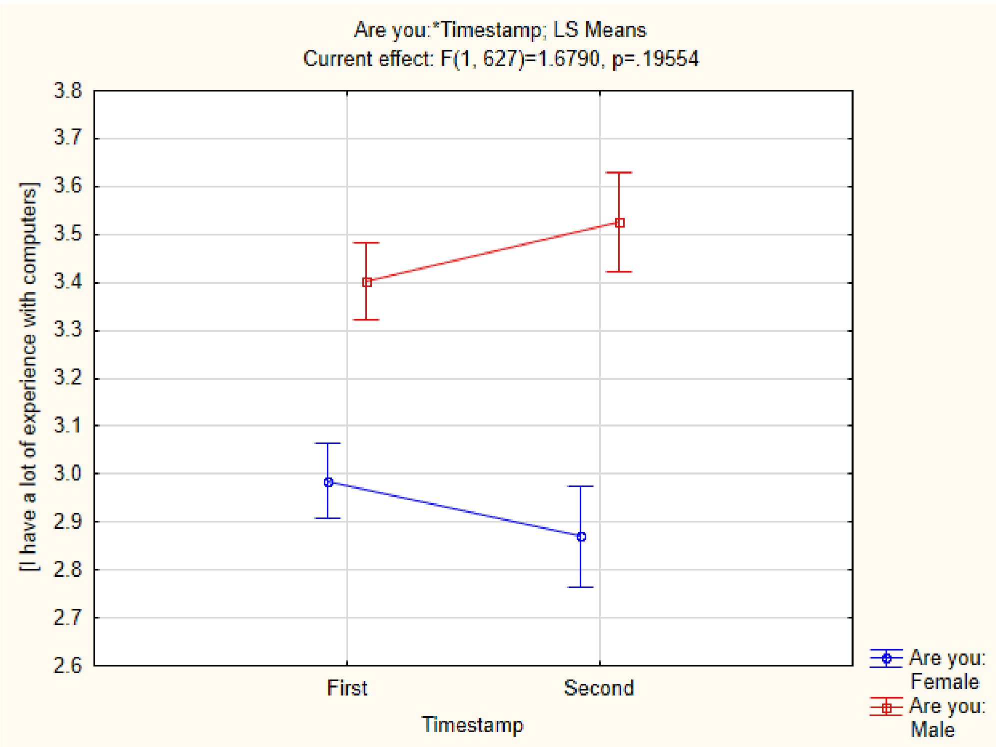

Figure A.2: Two-way ANOVA results for: I have a lot of experience with computers

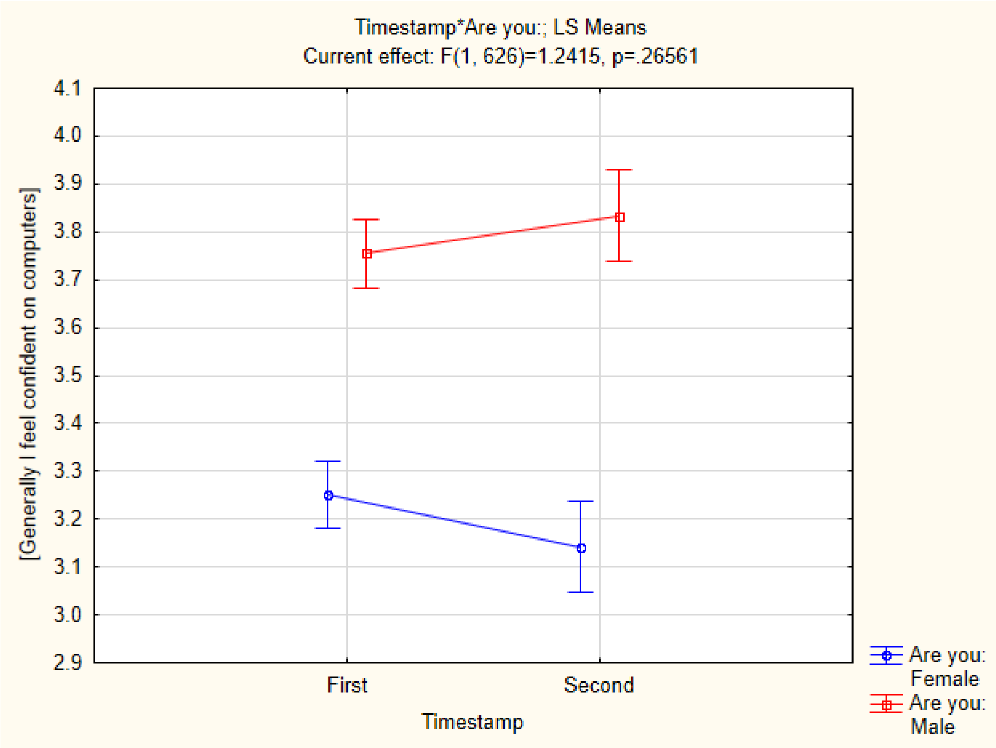

Figure A.3: Two-way ANOVA results for: Generally I feel confident on computers

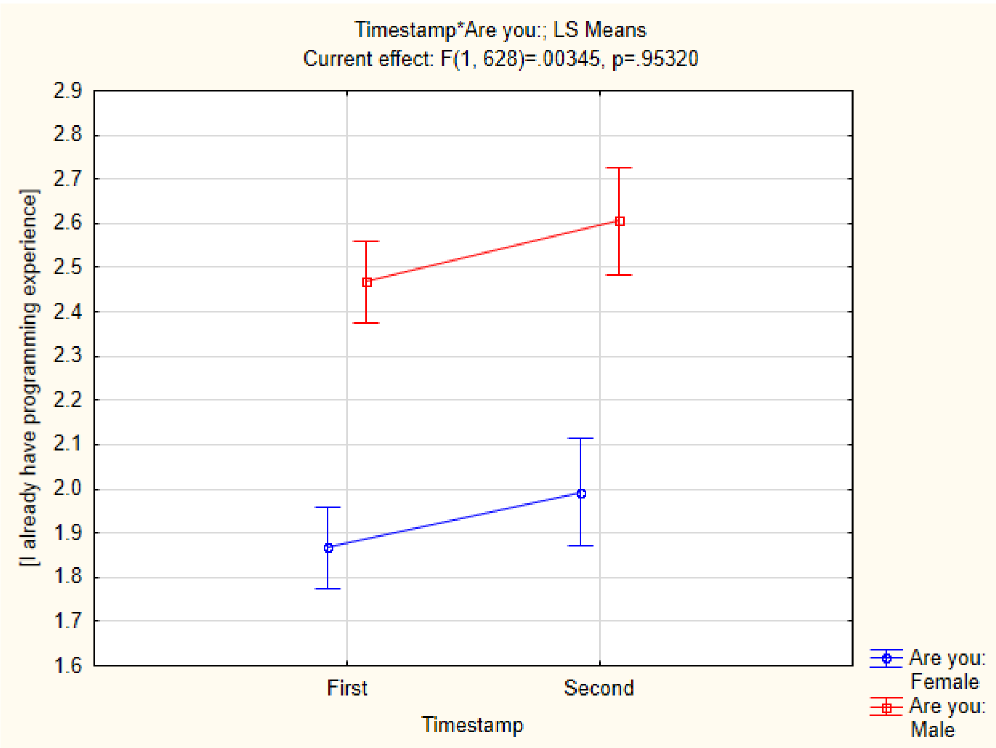

Figure A.4: Two-way ANOVA results for: I already have programming experience

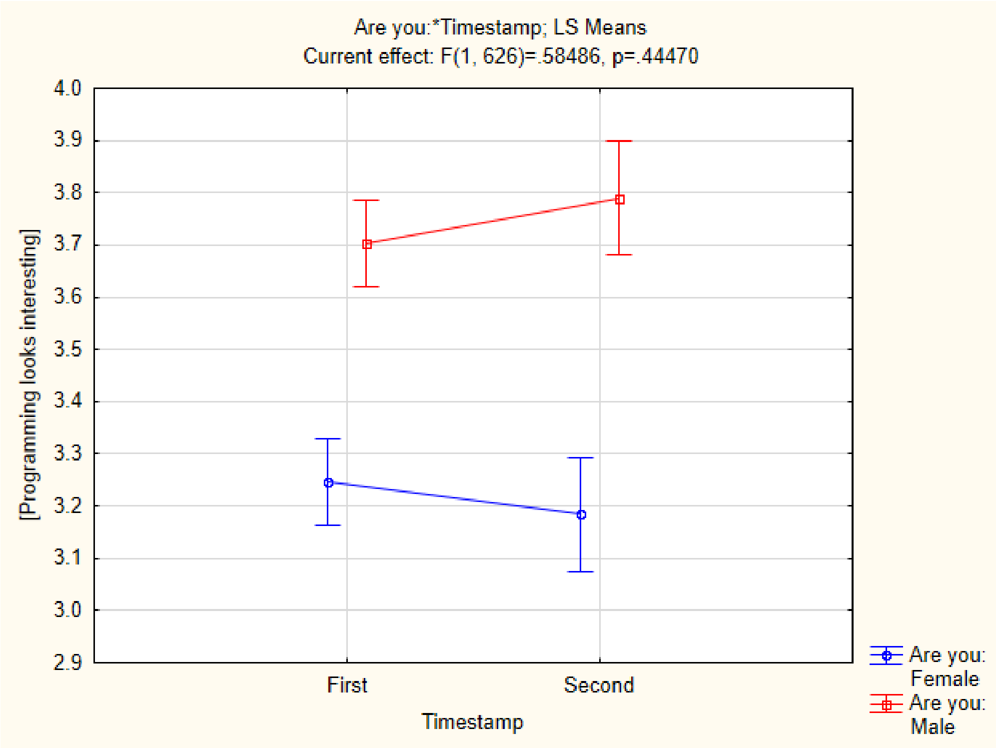

Figure A.5: Two-way ANOVA results for: Programming looks interesting

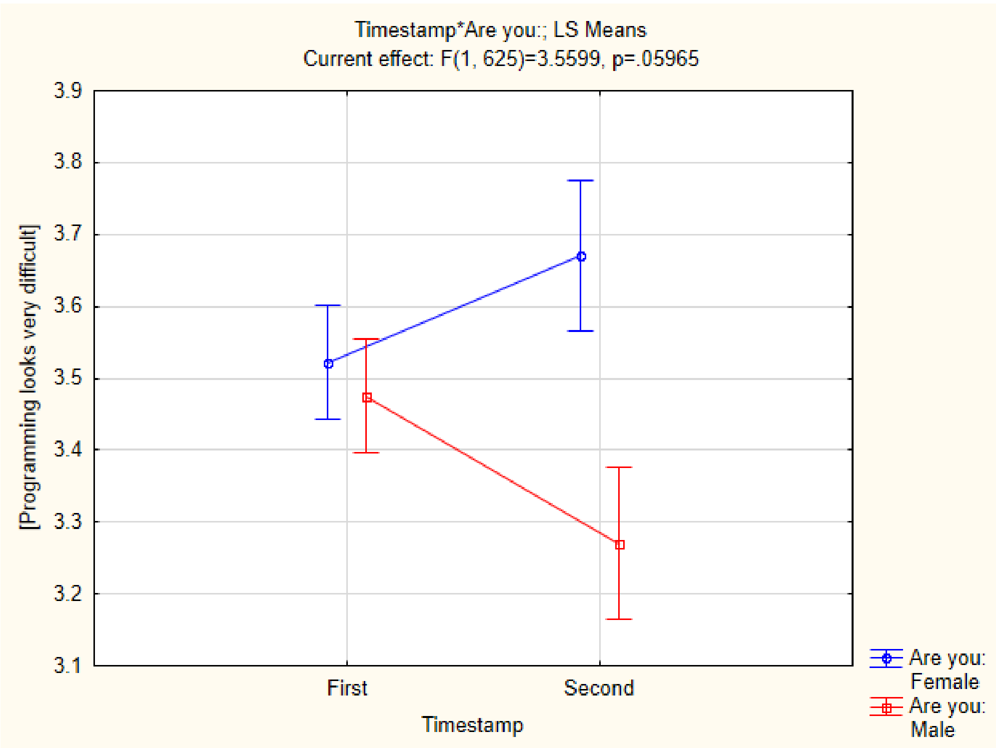

Figure A.6: Two-way ANOVA results for: Programming looks very difficult

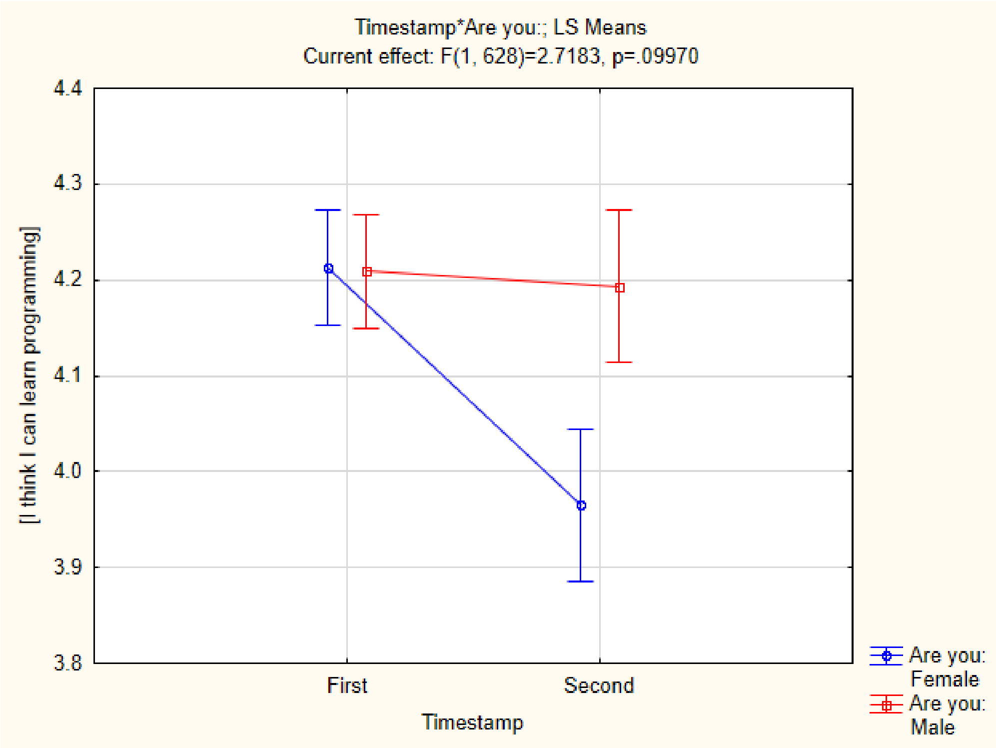

Figure A.7: Two-way ANOVA results for: I think I can learn programming

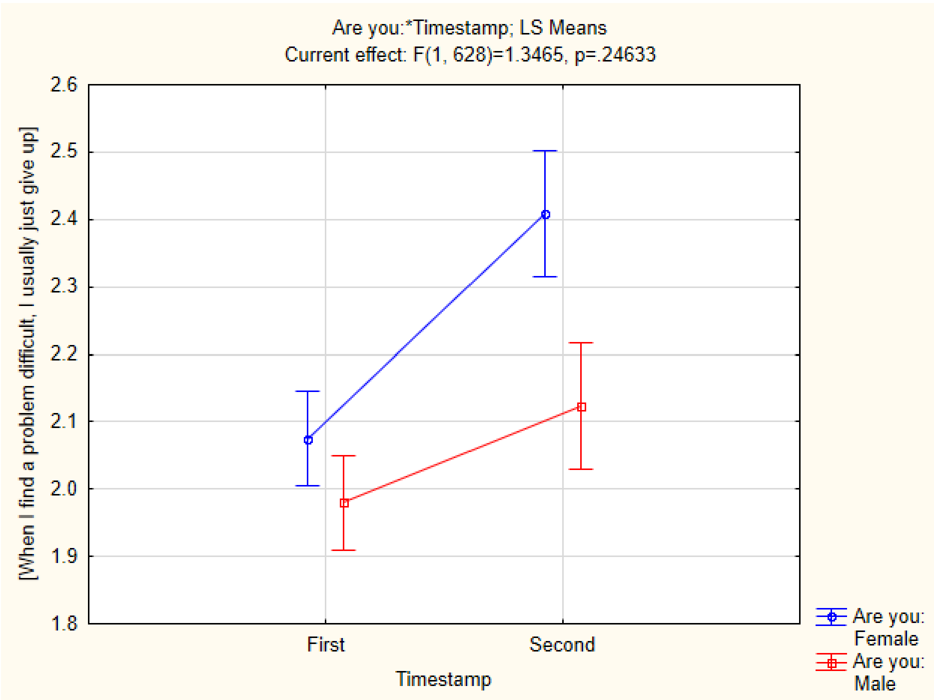

Figure A.8: Two-way ANOVA results for: When I find a problem difficult, I usually just give up

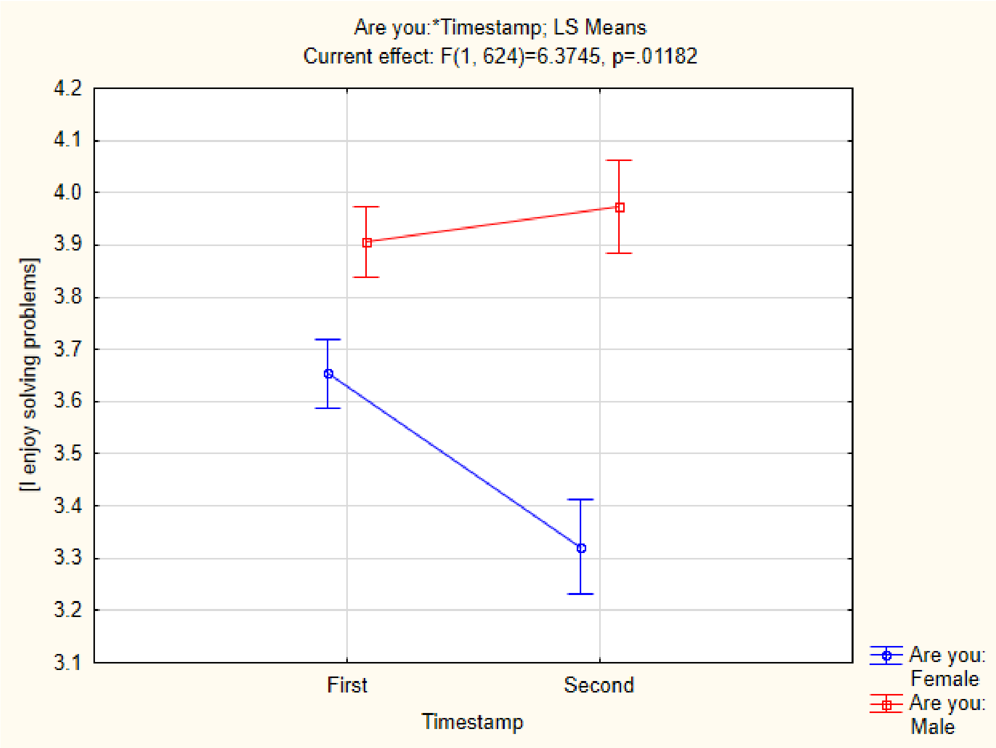

Figure A.9: Two-way ANOVA results for: I enjoy solving problems

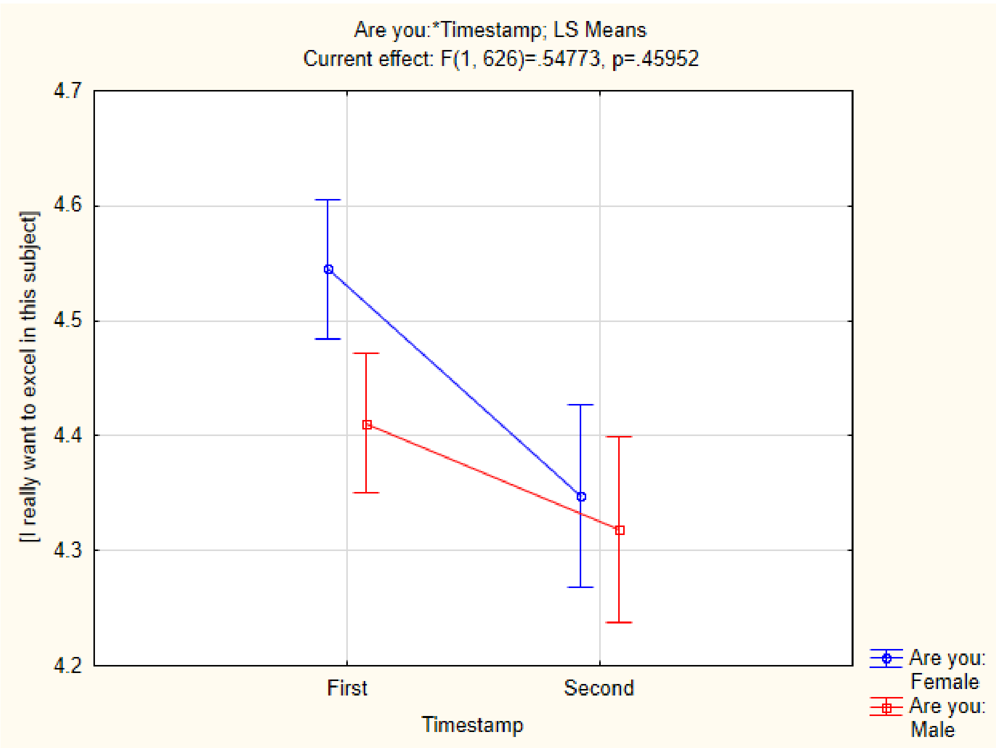

Figure A.10: Two-way ANOVA results for: I really want to excel in this subject

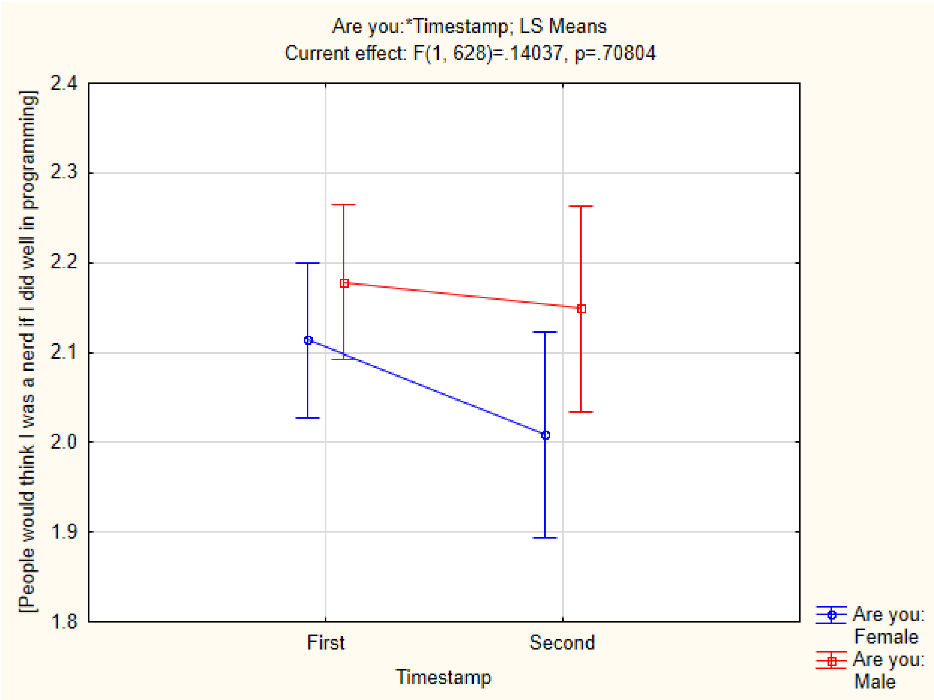

Figure A.11: Two-way ANOVA results for: People would think I was a nerd if I did well in programming

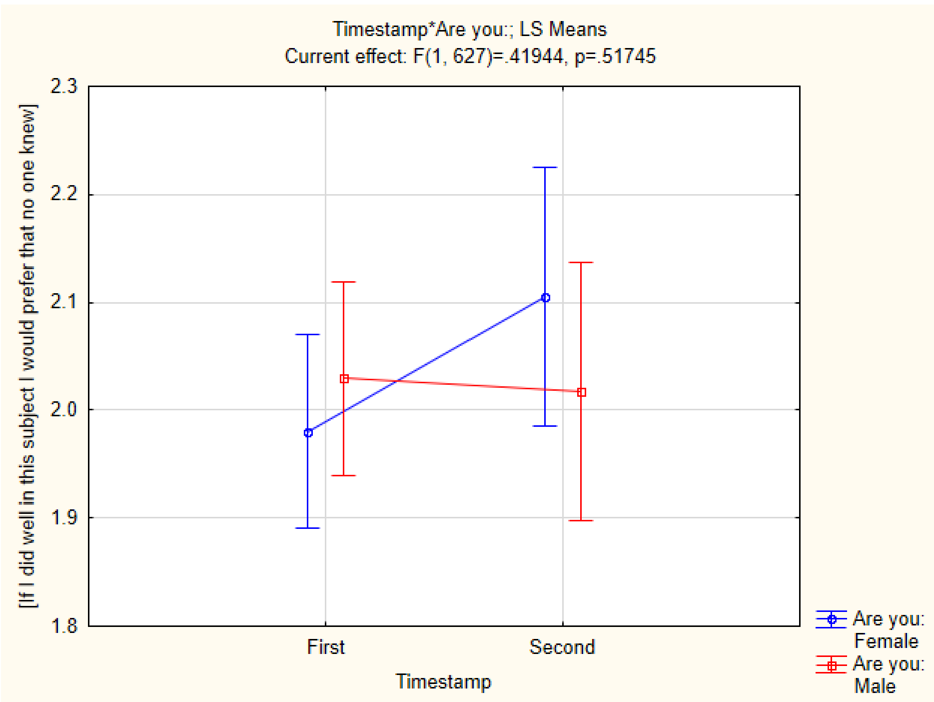

Figure A.12: Two-way ANOVA results for: If I did well in this subject I would prefer that no one knew

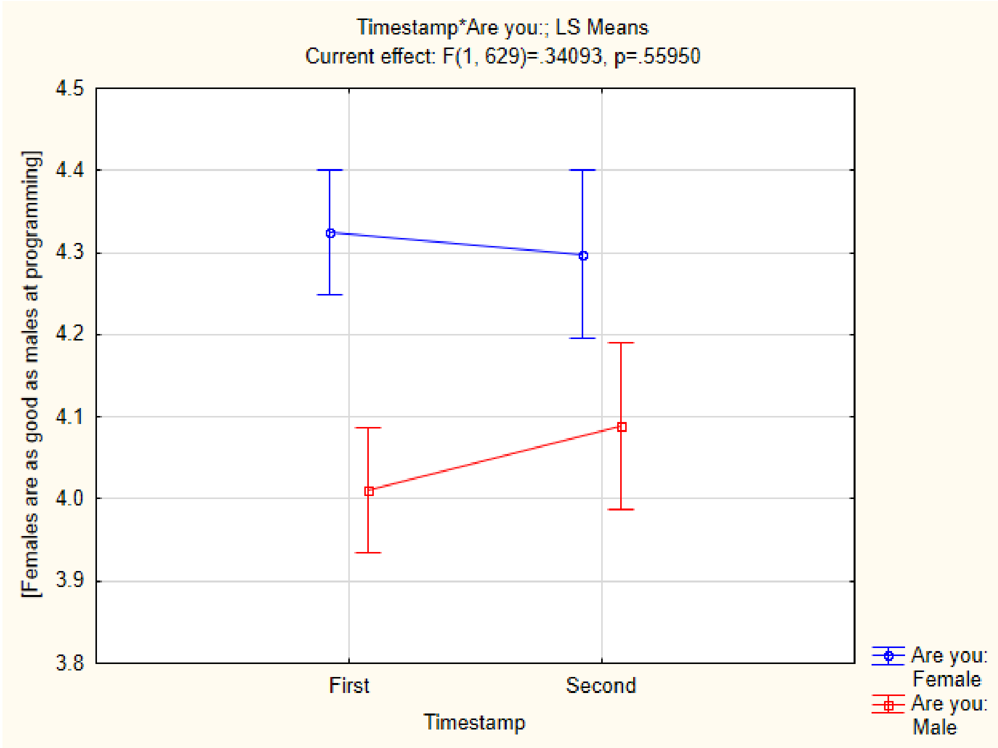

Figure A.13: Two-way ANOVA results for: Females are as good as males at programming

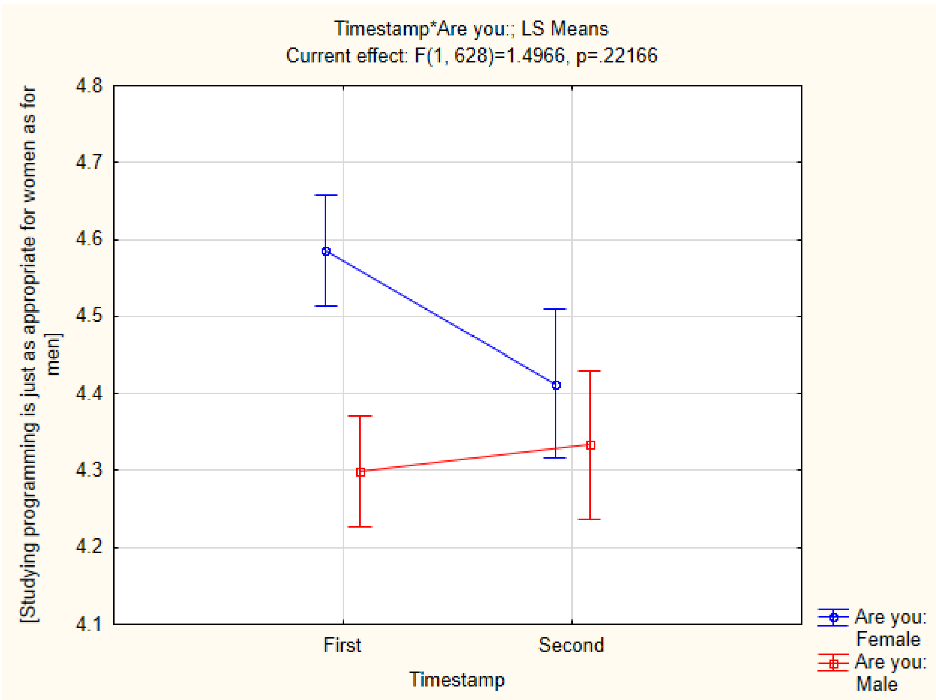

Figure A.14: Two-way ANOVA results for: Studying programming is just as appropriate for women as for men

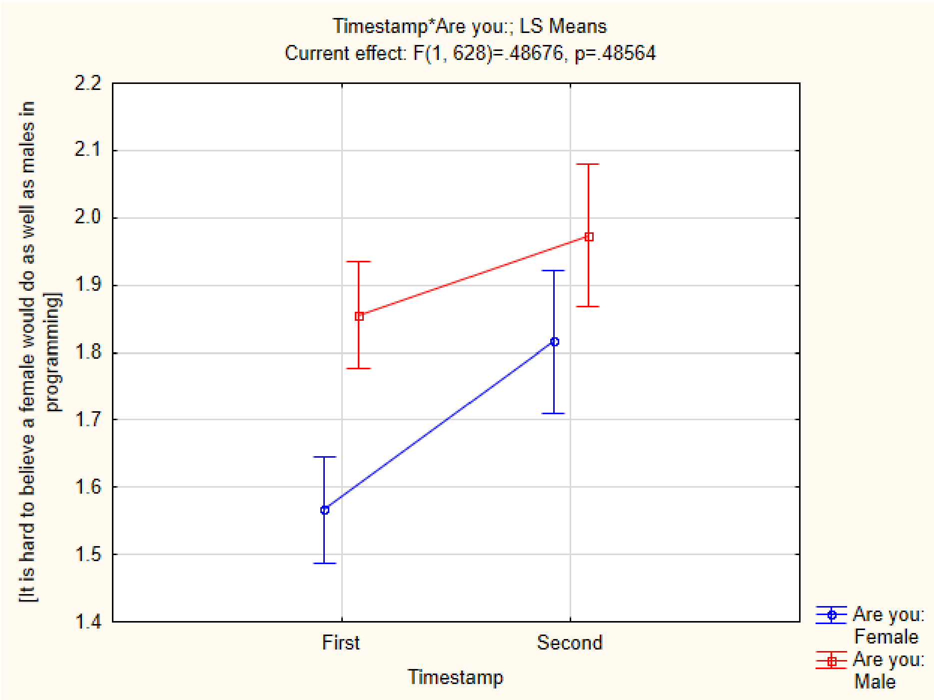

Figure A.15: Two-way ANOVA results for: It is hard to believe a female would do as well as males in programming

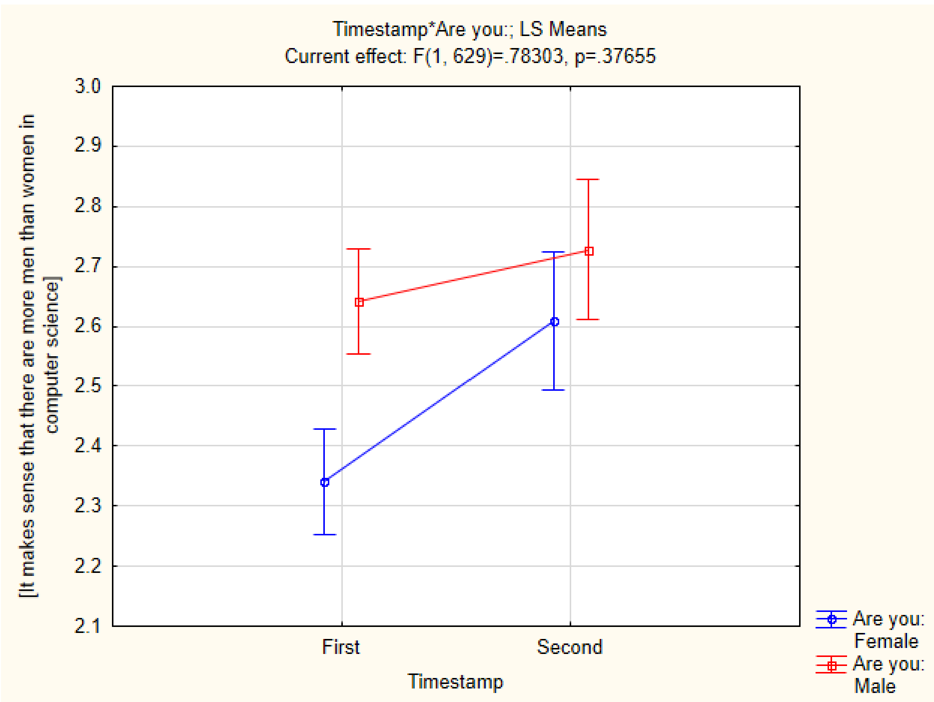

Figure A.16: Two-way ANOVA results for: It makes sense that there are more men than women in computer science

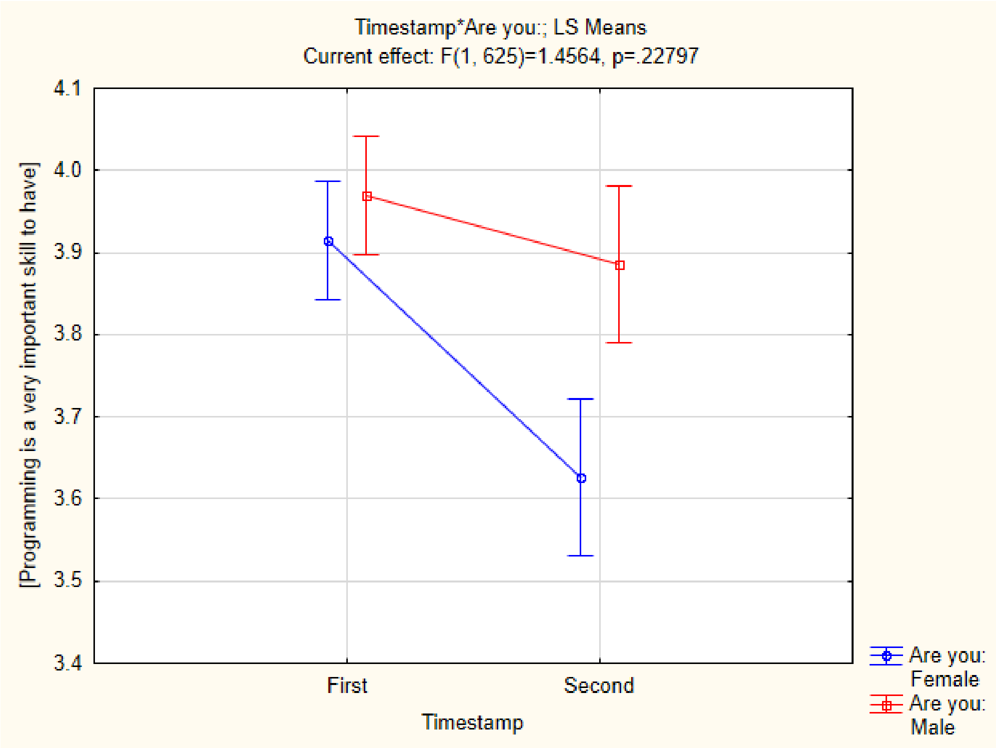

Figure A.17: Two-way ANOVA results for: Programming is a very important skill to have

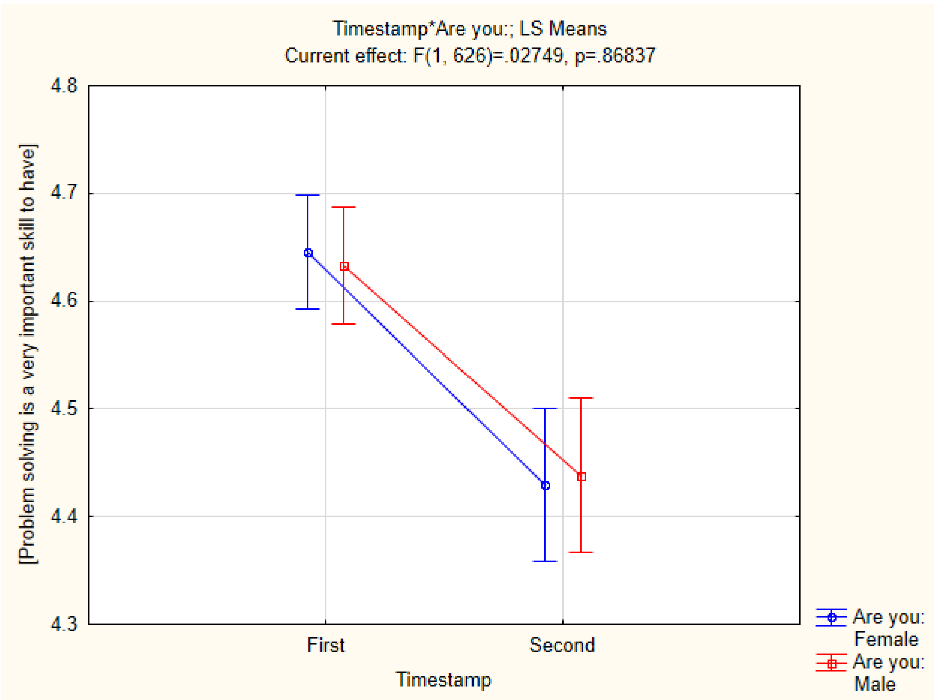

Figure A.18: Two-way ANOVA results for: Problem solving is a very important skill to have

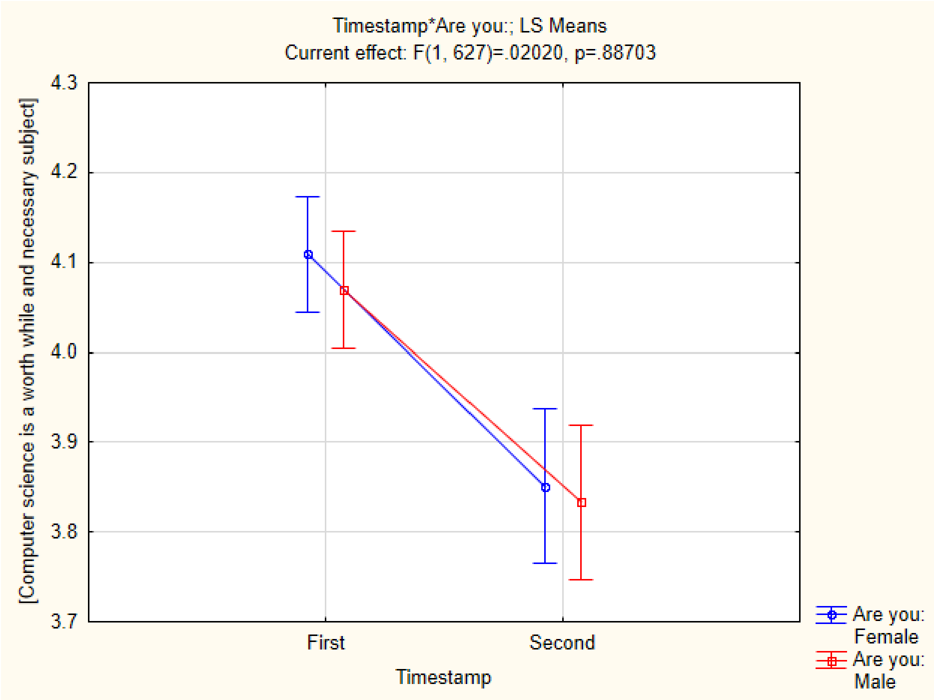

Figure A.19: Two-way ANOVA results for: Computer science is a worth while and necessary subject

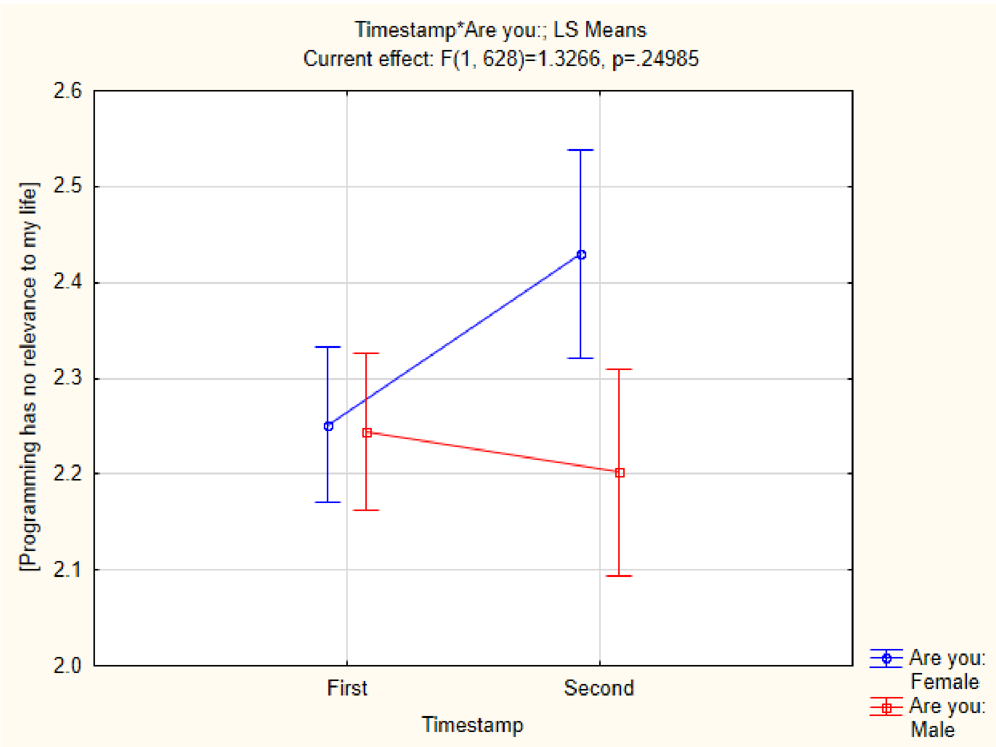

Figure A.20: Two-way ANOVA results for: Programming has no relevance to my life

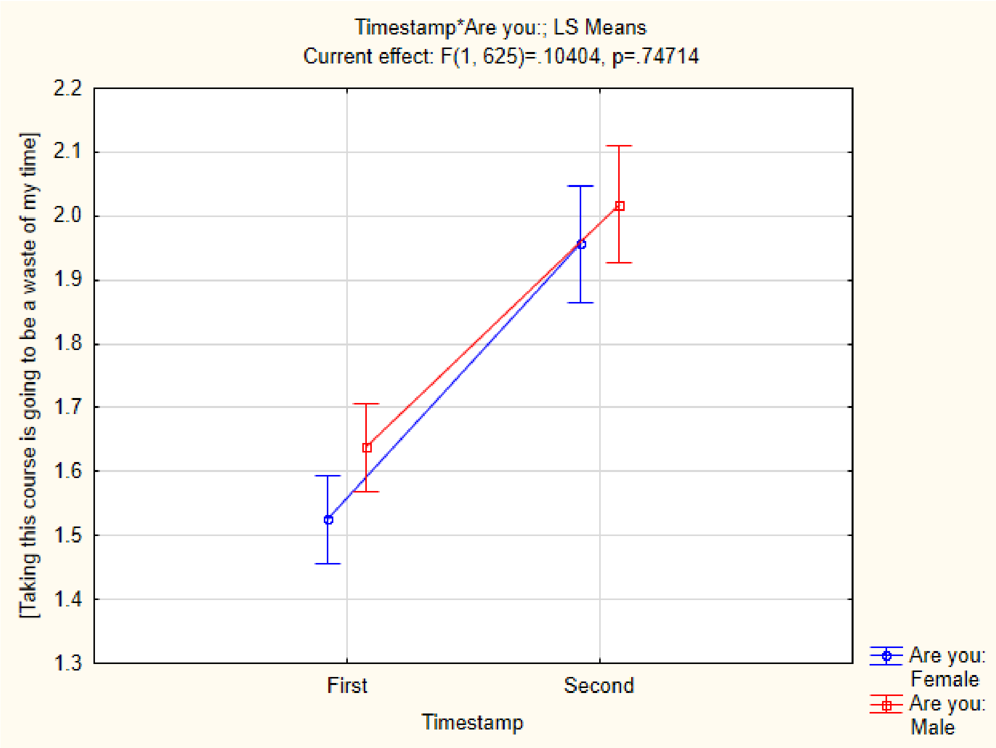

Figure A.21: Two-way ANOVA results for: Taking this course is going to be a waste of my time

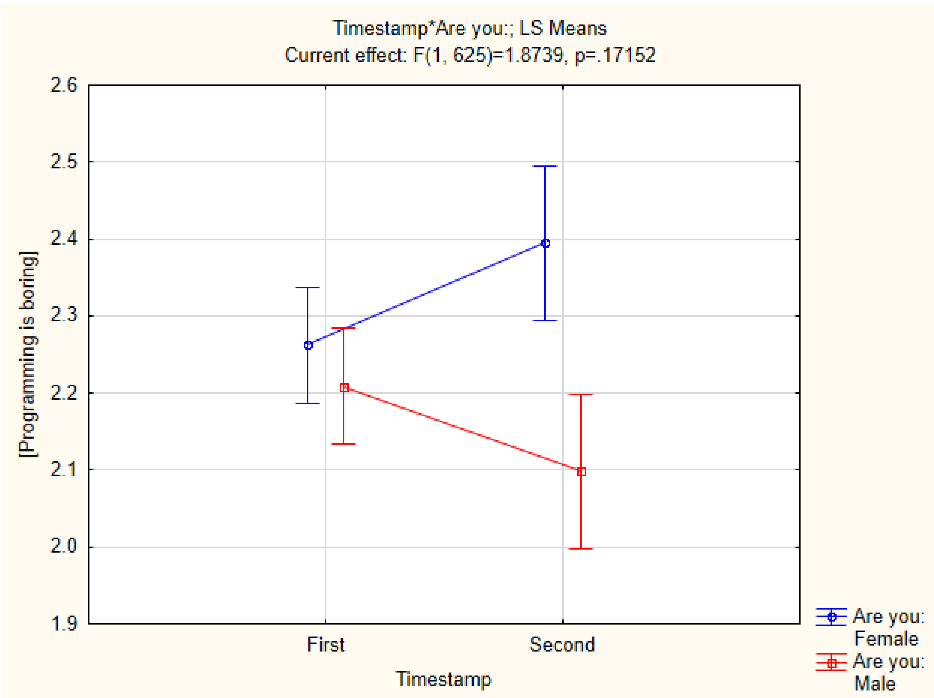

Figure A.22: Two-way ANOVA results for: Programming is boring

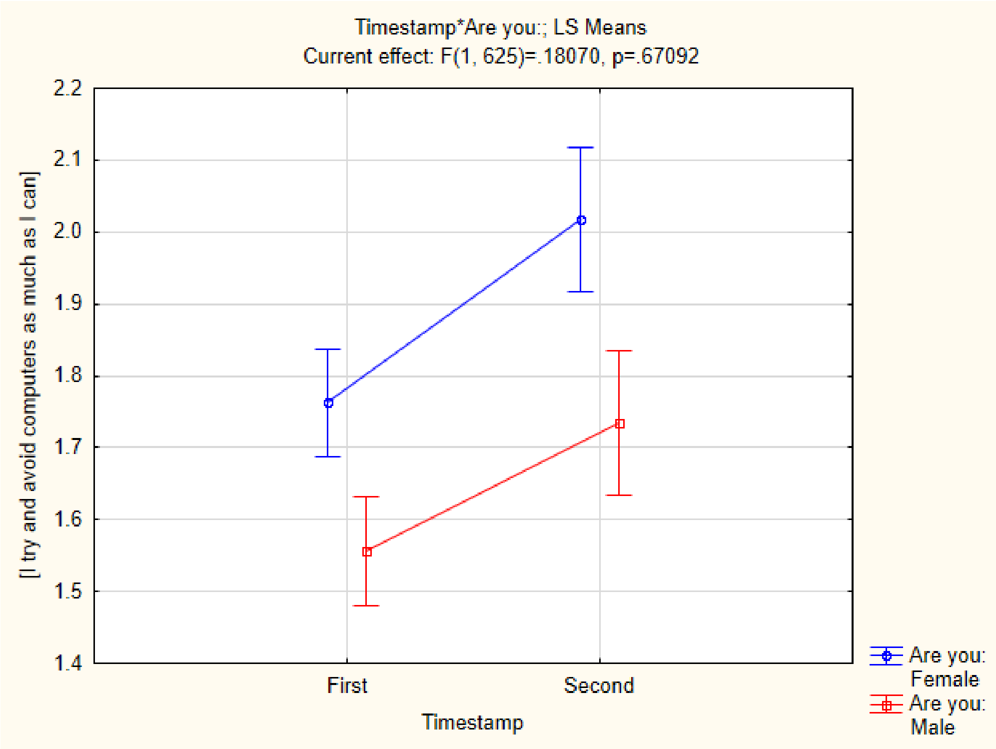

Figure A.23: Two-way ANOVA results for: I try and avoid computers as much as I can

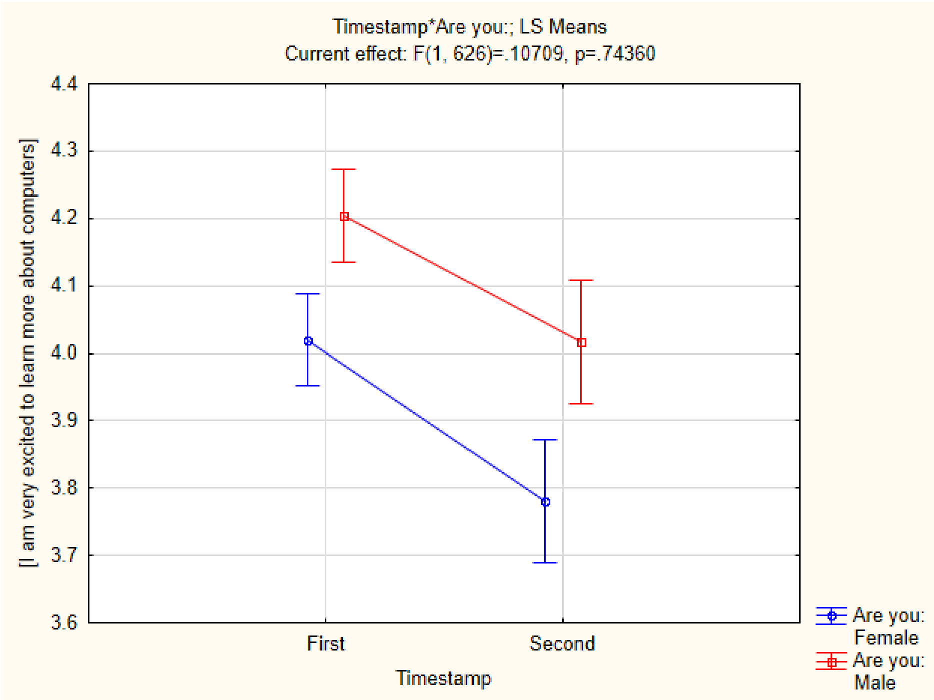

Figure A.24: Two-way ANOVA results for: I am very excited to learn more about computers

Appendix B

Interdepartmental survey questions

Honours Survey

This survey is to be used for research towards Mikha Zeffertt's Honours thesis for 2014. The research is being conducted to assess general opinions and perceptions about contextualized teaching for problem solving and computer science and the current gender divide evident in the field. The survey is anonymous and your contribution is greatly appreciated.
By proceeding to the next page and answering the questions you consent to participating in the research and allow me to use your responses as part of my thesis and any resulting publications. If you would like to follow up on the results of my research, please visit my project website at http://www.cs.ru.ac.za/research/g11z1718/

Contextualized learning and gendered teaching

Please answer the following questions and if you wish to comment or elaborate on a question, please do so in the text box below it.

1. **1. Are you…**

   Mark only one oval.

   - ( ) Male
   - ( ) Female

2. **2. Are you a…**

   Tick all that apply.

   - [ ] Student
   - [ ] Lecturer

3. **3. Which department do you fall under?**

   Tick all that apply.

   - [ ] Computer Science
   - [ ] Information Systems

4. **4. Do you feel the relevance of Computer Science and computational thinking is portrayed in CS lectures?**

   Mark only one oval.

   - ( ) Relevance is always apparent
   - ( ) Relevance is usually explained
   - ( ) Relevance is occasionally evident
   - ( ) Relevance is never brought up in lecturers
   - ( ) I don't know

5.

6. **5. Do you think there is a lack of computational thinking and problem solving skills amongst first year students?**
Mark only one oval.

- There is an extreme lack of skill
- Most students do not have these skills
- A few students struggle with them
- The skill level is really good
- I don't know

7.

8. **6. There is evidence to suggest teaching concepts while concurrently explaining real world contexts and applications of the data increases retention and interest in teaching material. Do you feel this is approach is used within the teaching of computer science, problem solving or computational thinking?**
Mark only one oval.

- Definitely
- Most probably
- Perhaps a little
- Not at all

9.

10. **7. What do you enjoy most about programming and development? Rate the following with 5 being very enjoyable and 1 being the least enjoyable.**

Mark only one oval per row.

	1	2	3	4	5
Problem solving					
Satisfaction of success					
Creativity					
Control					
Challenge					
Ownership					
Logic					

11.

12. **8. Science is often said to be taught in a highly abstracted and theoretical way, with little emphasis on real world application and favouring shorter questions without needing explainations. Do you think that the way computer science is taught in this style?**

Mark only one oval.

- No, not at all
- Perhaps slightly
- It is fairly biased
- Yes, it is very biased
- I haven't done computer science

13.

14. **9. Do you think changing the way Computer Science is taught would effect the difference in gender representation within the field?**

Mark only one oval.

- Yes
- No
- I don't know

15.

16. **10. Do you have any final comments about contextualized learning or the gender imbalance within the field of computer science?**

Powered by
Google Forms

Appendix C

Initial CSc 112 survey questions

CSc 112 Introductory survey

This survey is to be used for research towards Mikha Zeffertt's Honours thesis for 2014. The research is being conducted to assess the general attitudes of computer science 112 students towards this subject. The survey is anonymous and your contribution is greatly appreciated. Please answer all questions.
By proceeding to the next page and answering the questions you consent to participating in the research and allow me to use your responses as part of my thesis and any resulting publications. If you would like to follow up on the results of my research, please visit my project website at http://www.cs.ru.ac.za/research/g11z1718/

1. **Are you:**
   Mark only one oval.
   - ⬭ Male
   - ⬭ Female

2. **Are you in the extended studies program?**
   Mark only one oval.
   - ⬭ Yes
   - ⬭ No

3. **Why did you decide to take Computer Science 112? Select as many as are appropriate.**
   Tick all that apply.
   - [ ] You have to take it for your degree
   - [ ] You want to learn more about how to use a computer
   - [ ] You would like to learn basic programming skills
   - [ ] It seemed like an easy credit
   - [ ] A friend told you it was a fun course
   - [ ] You do not feel confident on a computer and want to learn more
   - [ ] It sounded interesting

4. **Rate each of the following statements based on how you feel about it:**
   There are no right or wrong answers. Don't spend to much time on a question, just respond with your first instinct.
   Mark only one oval per row.

	Strongly disagree	Disagree, but with reservations	Neutral, neither disagree or agree	Agree, but with reservations	Strongly agree
I plan to major in computer science					
I have a lot of experience with computers					
Generally I feel confident on computers					
I already have programming experience					
Programming looks interesting					
Programming looks very difficult					

5. *Mark only one oval per row.*

	Strongly disagree	Disagree, but with reservations	Neutral	Agree, but with reservations	Strongly agree
I think I can learn programming					
When I find a problem difficult, I usually just give up					
I enjoy solving problems					
I really want to excel in this subject					
People would think I was a nerd if I did well in programming					
If I did well in this subject I would prefer that no one knew					

6. *Mark only one oval per row.*

	Strongly disagree	Disagree, but with reservations	Neutral	Agree, but with reservations	Strongly agree
Females are as good as males at programming	⬭	⬭	⬭	⬭	⬭
Studying programming is just as appropriate for women as for men	⬭	⬭	⬭	⬭	⬭
It is hard to believe a female would do as well as males in programming	⬭	⬭	⬭	⬭	⬭
It makes sense that there are more men than women in computer science	⬭	⬭	⬭	⬭	⬭
Programming is a very important skill to have	⬭	⬭	⬭	⬭	⬭
Problem solving is a very important skill to have	⬭	⬭	⬭	⬭	⬭

7. *Mark only one oval per row.*

	Strongly disagree	Disagree, but with reservations	Neutral	Agree, but with reservations	Strongly agree
Computer science is a worth while and necessary subject	⬭	⬭	⬭	⬭	⬭
Programming has no relevance to my life	⬭	⬭	⬭	⬭	⬭
Taking this course is going to be a waste of my time	⬭	⬭	⬭	⬭	⬭
Programming is boring	⬭	⬭	⬭	⬭	⬭
I try and avoid computers as much as I can	⬭	⬭	⬭	⬭	⬭
I am very excited to learn more about computers	⬭	⬭	⬭	⬭	⬭

Appendix D

Follow up CSc 112 survey questions

CSc 112 Programming Logic Survey

This survey is to be used for research towards Mikha Zeffertt's Honours thesis for 2014. The research is being conducted to assess the general attitudes of computer science 112 students towards this subject and the computational thinking game in the Programming Logic Practical 1. It is a follow up of the survey conducted at the beginning of the term. The survey is anonymous and your contribution is greatly appreciated. It would be appreciated if you answered all questions.
By proceeding to the next page and answering the questions you consent to participating in the research and allow me to use your responses as part of my thesis and any resulting publications. If you would like to follow up on the results of my research, please visit my project website at http://www.cs.ru.ac.za/research/g11z1718/.

1. **Are you:**
   Mark only one oval.

   ◯ Male
   ◯ Female

2. **Are you in the extended studies program?**
   Mark only one oval.

   ◯ Yes
   ◯ No

3. **Rate each of the following statements based on how you feel about it:**
   There are no right or wrong answers. Don't spend to much time on a question, just respond with your first instinct.
   Mark only one oval per row.

	Strongly disagree	Disagree, but with reservations	Neutral	Agree, but with reservations	Strongly agree
I plan to major in Computer Science	◯	◯	◯	◯	◯
I have a lot of experience with computers	◯	◯	◯	◯	◯
Generally I feel confident on computers	◯	◯	◯	◯	◯
I already have programming experience	◯	◯	◯	◯	◯
Programming looks interesting	◯	◯	◯	◯	◯
Programming looks very difficult	◯	◯	◯	◯	◯

4. *Mark only one oval per row.*

	Strongly disagree	Disagree, but with reservations	Neutral	Agree, but with reservations	Strongly agree
I think I can learn programming	⬭	⬭	⬭	⬭	⬭
When I find a problem difficult, I usually just give up	⬭	⬭	⬭	⬭	⬭
I enjoy solving problems	⬭	⬭	⬭	⬭	⬭
I really want to excel in this subject	⬭	⬭	⬭	⬭	⬭
People would think I was a nerd if I did well in programming	⬭	⬭	⬭	⬭	⬭
If I did well in this subject I would prefer that no one knew	⬭	⬭	⬭	⬭	⬭

5. *Mark only one oval per row.*

	Strongly disagree	Disagree, but with reservations	Neutral	Agree, but with reservations	Strongly agree
Females are as good as males at programming	⬭	⬭	⬭	⬭	⬭
Studying programming is just as appropriate for women as for men	⬭	⬭	⬭	⬭	⬭
It is hard to believe a female would do as well as males in programming	⬭	⬭	⬭	⬭	⬭
It makes sense that there are more men than women in Computer Science	⬭	⬭	⬭	⬭	⬭
Programming is a very important skill to have	⬭	⬭	⬭	⬭	⬭
Problem solving is a very important skill to have	⬭	⬭	⬭	⬭	⬭

6. *Mark only one oval per row.*

	Strongly disagree	Disagree, but with reservations	Neutral	Agree, but with reservations	Strongly agree
Computer Science is a worth while and necessary subject					
Programming has no relevance to my life					
Taking this course is going to be a waste of my time					
Programming is boring					
I try and avoid computers as much as I can					
I am very excited to learn more about computers					

7. *Mark only one oval per row.*

	Strongly disagree	Disagree, but with reservations	Neutral	Agree, but with reservations	Strongly agree
I will use what I learnt in the programming course in my daily life					
I continue to work at a problem when I can't immediately solve it					
Once I start working on a problem I find it difficult to stop					
The challenge of solving problems is something that I find really interesting					
I don't understand how people can spend so much time on writing programs and enjoy it					
I would rather have someone give me the solution to a difficult problem than work it out myself					

8. **Did you play the game in last week's practical?**
Mark only one oval.

- Yes *Skip to question 9.*
- No *Stop filling out this form.*

Computational Thinking Game

9. **Did you enjoy playing the game?**
Mark only one oval.

- Yes, I greatly enjoyed it
- I enjoyed it a lot but I would rather be doing something else
- It was ok
- No, it was pretty boring and felt pointless

10. **Did you feel that the game explained the concepts clearly?**
Mark only one oval.

- All the concepts made sense
- I understood most of them, but some weren't clear
- I understood a few but mostly I was just guessing my way through
- They made no sense at all

11. **Did you like being able to choose your own character?**
Mark only one oval.

- Yes, it made me care about my little character more
- It didn't really make a difference to the game
- No, I thought it was a pointless extra

12. **Did you choose a character that resembled yourself (race or gender)?**
Mark only one oval.

- Yes
- No
- I can't remember

13. **Do you think using games for explaining how concepts work helped you learn?**
Mark only one oval.

- Definitely! It makes it really easy
- Neutral, they help but lectures and regular pracs help too
- No. The game was completely unhelpful

14. **When playing, did you rely more on reading the instructions or figuring out how to do things on your own?**
Mark only one oval.

- I only followed the instructions
- I read the instructions, but not too carefully.
- I looked at them but hardly read them.
- They were a waste of time. I skipped through as fast as possible.

15. **Do you wish some more of the computer science modules used games for teaching?**
Mark only one oval.

- Yes
- Maybe a few more should
- No

16. **The game allowed you to try out the concepts as you were learning them. Do you like learning in context like this, where a concepts use is explained along side how it works?**
Mark only one oval.

- Yes, I think it makes more sense this way
- Neutral, it is nice sometimes but often it is unnecessary
- No. I think it takes too much time and I would rather just learn the pure concepts

17. **Did playing the game help increase your confidence in being able to problem solve?**
Mark only one oval.

- Definitely, I feel much better about solving problems
- Neutral, it didn't increase of decrease my self-confidence in problem solving
- No, I think it made my confidence worse

18. **When you got stuck in the game, how did you handle it?**
Mark only one oval.

- I just tried a lot of different things until something worked.
- I asked for help.
- I tried but when I couldn't get it, I went back to the given instructions.
- I did not get stuck.

19. **Do the normal practicals for Computer Science 112 increase your self confidence in being able to problem solve?**

Mark only one oval.

- Definitely, I am much more confident because of the pracs
- Neutral, the pracs do not effect my confidence
- No, I think they make my confidence worse

20. **Did this game increase your interest in Computer Science?**

Mark only one oval.

- Yes
- Neutral
- No

21. **If you have any comments on any of the above questions, on the game in general or on the course, please state them below. This is optional but encouraged.**

Powered by Google Forms