Computer Science Honours

Literature Review

An investigation into the benefits of contextualised learning in Computer Science

Mikha Zeffertt

Department of Computer Science, Rhodes University

Supervisor:
Mici Halse

Supervisor:
Hannah Thinyane

May 30, 2014

Contents

1 Introduction

University enrolment rates for computer science have dramatically decreased since 2000 (Bayliss, 2009, Carbonaro *et al.*, 2010, Carter, 2006, Muratet *et al.*, 2009), with some universities reporting that more than 50% of students that begin studying computer science abandon the field (Muratet *et al.*, 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 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 will explore 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 will be described and evaluated. There will also be an exploration of what reasonings currently exist regarding the state of education within computer science. In particular, the subject of computational thinking will be a focus area within introductory computer science.

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 will provide 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, will increase 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. Due 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 one year, and was compared against a year taught by a different lecturer. The results are not conclusive but show encouraging progress while using contextualized teaching, with an average mark increase of over ten percent.

In 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, is very different from the one 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).

2.1 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 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. Due 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 in order 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).

2.3 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 *et al.* (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).

2.4 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.

2.5 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. Due to 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.

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 (Jorgensen *et al.*, 2013, 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, Repenning *et al.*, 2010). If students have fun while using the application, they become encouraged to play more than the required amount (Chang *et al.*, 2012, Repenning *et al.*, 2010). 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.

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. Due 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). Due 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.

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 at 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 regards 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. In order 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).

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 *et al.* (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).

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.

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 (Bombardieri, 2005, 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. 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.

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 *et al.*, 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.

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 multiplayer 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 *et al.* (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-practise 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.

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 (Casey *et al.*, 1997, Bandalos *et al.*, 1995). However, it must be acknowledged that males have a much wider distribution across mark ranges than females (Casey *et al.*, 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.

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).

5 Conclusion

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).

References

Anon. 2009. Gender Differences in Science Achievement. Online. Available from: `https://www.engr.psu.edu/AWE/misc/ARPs/ARP_InfoSheet_Science.pdf<Accessed26/02/2014>`.

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.

Bombardieri, M. 2005. In compter science, a growing gender gap: Women shunning a field once seen as welcoming. Online. Available from: `http://boston.com/news/education/higher/articles/2005/12/18/in_computer_science_a_growing_gender_gap/?page=full`.

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. *In: 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.

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. *In: 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.

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. *Mathematics and gender*. Teachers College Press. Chap. Spatial skills, gender, and mathematics., pages 27–59.

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.