Digital Games & Computational Thinking in Pre-Service Teacher Education

by

Corbett Raymond Walter Artym

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Education

in Technology in Education Specialization

Department of Educational Psychology
University of Alberta

Abstract

The thesis consists of two papers exploring the area of digital game construction in pre-service teacher education. The first paper details the analysis of 166 pre-service teachers' experiences constructing a digital game in the *Scratch* development environment (MIT, 2009). Pre-service teachers (64% male, 36% female) had no previous digital game creation experience and self-selected into an elective educational technology course that included this game construction activity. The purpose of this research was twofold: A) to find if pre-service teachers have any predisposition to digital game creation relating to genre, gender, and previous time spent playing digital games or using social media, and B) to quantitatively assess the computational thinking and game design skills demonstrated in the game they create. In the first paper, the games were classified into nine genre categories, identified from the literature, and their differences were compared. Results indicate a significant quadratic relationship between genders on previous time spent game playing across the different age ranges that were explored (males played more). Both genders reported playing fewer hours of games in elementary school and university, but more in junior and senior high school. There was also an increase in usage of social media as these pre-service teachers progressed from elementary school to university. As a whole, pre-service teachers are significantly more likely to construct action games with non-violent genres. However, when gender is a factor, males are significantly more likely to create violent action games, whereas there was no significance when testing the preferred game genre created by females. In the second paper, the Quality Practices of Game Design Survey was developed to measure the skills pre-service teachers demonstrated in their created game. A comprehensive review of the literature identified 28 key skills which can be grouped into seven categories: Problem Solving, Computational Thinking, Customization of Player Experience, Player

Interaction, Player Immersion, Player Motivation, and Interface Usability. A purposeful sample was selected (40 games) and used to evaluate the survey instrument. Frequencies were found in evaluations and items were compared in the form of a correlational matrix. Overall, the set of video games built by the pre-service teachers indicate that they have a partial, but not complete, awareness of computational and game design principles. This thesis may be valuable in motivating interventions to compensate for potential game design predispositions and for developing an assessment tool for computational thinking and game design skills outside of *Scratch*.

Keywords: digital games, video games, computational thinking, game design, gender, genre, pre-service teacher education, social media usage experience, video game play history, Scratch, learning, technology

Preface

This thesis is an original work by Corbett Artym. The research project, of which this thesis is a part, received research ethics approval from the University of Alberta Research Ethics Board, Project Name “Video Games in Education”, No. Pro00012268, March 17, 2010 –March 15, 2012, and “Video Game-Building as a Teaching Tool”, No. Pro00030372, April 14, 2013-April 14, 2014. At the time of defense, no part of this thesis has been previously published.

Some of the research conducted for this thesis forms part of a research collaboration, led by Professor Patricia Boechler, with Professor Mike Carbonaro at the University of Alberta, and with the assistance of Erik deJong, Anthony You, Cody Steinke, Connie Yuen, and Wayne Thomas. The Genre Classification scheme in the first paper and *Quality Practices of Game Design Survey* (QPGDS) referred to in the second paper were designed by Corbett Artym, with the assistance of Michelle Killoran and Professor Mike Carbonaro.

Dedication

This thesis is dedicated to my parents who have motivated me every step along the way. Mom and Dad, thank you for providing me with food or whatever assistance I needed. This was most valuable when I have been too distracted with school to remember that I need to sustain my own needs. I also dedicate this thesis to my sister, Bailey, who I requested be returned to the hospital when she was first brought home, but ended up being one of my best friends.

Acknowledgements

My master's thesis has been a journey made possible through the support of many.

My supervisor, Dr. Mike Carbonaro, has been an invaluable guide on this journey. He has provided me with many constructive opportunities helping me to develop both my research and teaching aptitudes. Without any harassment, Mike has made time to meet with me, call me, and respond to emails at all times of the day. Thank you for being patient and accommodating when met with my ambition and stubbornness. I appreciate everything he has taught me throughout my program.

I would like to thank Michelle Killoran for all her support and contributions to these projects. Michelle has been irreplaceable friend and colleague throughout both my undergraduate and graduate degrees. I look forward to the two of us continuing to assist each other while teaching Computing Science.

I would like to express my appreciation to Man-Wai Chu for her assistance with the statistics on this project, which was not one of my strengths. Thank you for articulating these atrocities in a way that even I can understand.

Thank you to Janet Le for looking over my papers throughout my program and providing helpful feedback.

I would also like to thank the members of my defense committee, Dr. Patricia Boechler and Dr. George Buck, not only for their suggestions, and comments, but also for all their work as Lab Director and Graduate Coordinator, respectively during my program. Both have also been instrumental in improving my writing abilities.

A memorable part of my program was teaching EDIT 486: Interactive Multimedia. I would like to thank all of my students, and incidentally the participants for the studies explored in my papers, for making my time in this master's program so much fun.

Finally, I would like to thank my overwhelmingly wonderful friends and family, who I cherish knowing. Each one of you means a lot to me; I am a social creature and you make me feel loved. Time is a limited resource; thank you for making time for me.

Table of Contents

List of Tables

Evaluating Computational Thinking in Digital Games Constructed by Pre-service Teachers

Appendix

List of Figures

Pre-service Teachers Constructing Digital Games: What Genre do they create?

Introduction

The thesis consists of two papers exploring the area of digital game construction in pre-service teacher education. The first paper investigates if pre-service teachers have any predisposition to digital game creation relating to genre, gender, and previous time spent playing digital games or using social media. The second paper quantitatively assesses the computational thinking and game design skills demonstrated in the games pre-service teachers create.

Digital game construction is a viable problem space for pre-service teachers to develop quality pedagogical practice and to further engage their own students through focused goals, challenging tasks, clear and compelling standards, protection from adverse consequences for initial failures, affirmation of performance, affiliation with others, novelty and variety, choice, and authenticity. The basis for the research work in this thesis is succinctly described in the following quotes from Lim (2008) and Gershenfeld (2011), for paper one and paper two respectively:

> If educators design learning experiences based solely on their own vision, goals and circumstances, they may be merely imposing their set of values upon their students; engaged learning is unlikely to happen in such an environment. It is only when students are empowered to take charge of their own learning by co-designing their learning experiences with teachers and other students that they are more likely to engage in their learning process. One way of doing so is to allow students to be the designers of their own computer games based on their own interpretations of the school curriculum (Lim, 2008, p. 1002).
>
> Designing a digital game requires one to think analytically and holistically about games as systems, to experiment and test out theories, to solve problems, to think critically, and to effectively create and collaborate with peers and mentors. These are all skills that will be needed in a twenty-first century where virtually every job will involve navigating a complex, ever changing, digitally networked global landscape and where many of the future jobs have yet to be invented (Gershenfeld, 2011, p. 55).

When integrating technology, to support and enhance teaching and learning, teachers need to be aware of: a) their own personal predispositions toward the technology they are using; and b) how they might evaluate the learning of the technology integration activity. The first paper presented explores whether pre-service teachers have a predisposition to creating a certain genre of game, and if this was unique to a specific gender, the amount of time they engaged in video game play and social media usage. Pre-service teachers (166) each constructed their own video game of their choice. These games were classified into nine genre categories, identified from the literature, and their differences were compared. The games were also analyzed with respect to gender, and the relationship to previous time spent playing games and using social media. Results indicate a significant difference between the genders on previous time spent game playing across different age ranges (males played more); interestingly the significant relationship is quadratic; both males and females reported playing less in elementary and university, but more in junior and senior high school. There was also an increase in usage of social media as these pre-service teachers progressed from elementary to university. As a whole, pre-service teachers are significantly more likely to construct action games with non-violent genres. However, when gender is a factor, males are significantly more likely to create violent action games.

Digital game design and the act of programming can help pre-service teachers and their students develop computational thinking skills. Wing (2006) outlined computational thinking specifying that "computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (p. 33), further emphasizing that computational thinking is a skill that everyone should have. The second paper identifies qualities of game design and computational thinking through a comprehensive review of the literature, and then quantitatively outlines these qualities in the *Quality Practices of Game Design Survey* tool to assess them. The QPGDS consists of 28 items grouped into 7 distinct categories: Problem Solving, Computational Thinking, Customization of

Player Experience, Player Interaction, Player Immersion, Player Motivation, and Interface Usability. A purposeful sample was used to evaluate the instrument. Frequencies found in evaluations are represented in pie and bar graphs; in addition, items were compared in the form of a correlational matrix. Overall, the set of video games built by the pre-service teachers indicate that they have a partial, but not complete, awareness of computational and game design principles.

With the introduction of new user-friendly tools for game creation, pre-service teachers are able to acquire twenty-first century skills before passing these onto their students who will be the professionals of tomorrow. More specifically, video game design and construction offers an interesting and exciting platform for the development of computational thinking skills.

References

Gershenfeld, A. (2011). Leveling up from player to designer: engaging and empowering youth through making video games. *Knowledge Quest*, *40*(1), 54-59. Retrieved from http://www.editlib.org/p/111220/

Lim, C. P. (2008). Spirit of the game; empowering students as designers in schools. *British Journal of Educational Technology*. doi: 10.1111/j.1467-8535.2008.00823_1.x

Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi: 10.1145/1118178.1118215Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi: 10.1145/1118178.1118215

Pre-service Teachers Constructing Digital Games: What Genre do they create?

Abstract

In this exploratory study, 166 pre-service teachers each constructed a video game of their choice using the *Scratch* development environment (MIT, 2009). The pre-service teachers (64% male, 36% female) had no previous video game creation experience and self-selected into an elective educational technology course that included this game construction activity. Data collected included 166 games (7% educational, 93% entertainment) and survey information on previous game playing and social media experience. The 166 games were classified into nine genre categories, identified from the literature, and their differences were compared. The games were also analyzed with respect to gender, as well as previous time spent playing games or using social media. Results indicate a significant difference between males and females on previous time spent game playing across different age ranges, with males indicating they played more. This significant relationship is quadratic (i.e., both males and females reported playing less in grades 1-6 and university, but more in grades 7-9 and high school). There were no differences with respect to social media. Overall, pre-service teachers are significantly more likely to construct action games with non-violent genres. When gender is a factor, males are significantly more likely to create violent action games. This research may be valuable in motivating interventions to compensate for potential game design predispositions.

Keywords: digital games, video games, game design, gender, genre, pre-service teacher education, social media usage experience, video game play history, Scratch, learning, technology

Technology pervades people's personal and professional lives, and pre-service teacher education is no exception (Williams, Foulger, & Wetzel, 2009). There is ample evidence to suggest that technology integration into classroom practice does have a positive effect on learning and attitudes, but there is a wide variation of effects due to educational context, content domain, and pedagogical approach, all of which can significantly modify the outcomes (Schmid et al., 2014; Tamin, Bernard, Borokhovshi, Abrami, & Schmid, 2011). One area of technology integration that is receiving increasingly more attention is the use digital games to support learning when students are players or modifiers of existing video games (Gee, 2013; Schrader, Archambault, & Oh-Young, 2011; Tobias & Fletcher, 2011; Whitton, 2014). For example, the recently released Coursera Massive Open Online Course (MOOC), *Video games and Learning,* by Steinkuehler and Squire (2014), offers students a chance to "examine the inherent tensions between contemporary youth culture and traditional education and new developments in games for learning that promise to help bridge that growing divide." Many resources and curriculum materials are also being developed to support teachers in using digital games in their classrooms (Shapiro et al., 2014; Bos, Wilder, Cook, & O'Donnell, 2014; Smith & Neumann, 2014).

Arguments for how digital games should be integrated into classroom practice have primarily focused on students as *players* of existing digital games and not as *builders* of their own game; however, there is increasing evidence to indicate that positive learner outcomes can be achieved when students are both designers and builders of digital games (Carbonaro et al., 2008; Good, 2011; Denner, Wener, & Ortiz, 2011; Robertson & Howells, 2008; Vos, van der Meijden, & Denessen, 2011). Prensky (2008) advocates that when students are placed in the role of designer and constructor of video games, there is increased student engagement and an enhancement of the school curriculum.

The notion that student learning can benefit from the building of personal, sharable, and computational artefacts has its roots in Papert's (1971; 1980; 1991) early work with the Logo program language and the development of his 'constructionist' learning theory. Papert's argument is that students' learn most effectively when they construct/build a tangible item (e.g., book, robot, or game, etc.). Penner (2000) makes a strong argument that a student's building of computational models (e.g. digital games) can result in a powerful form of learning and understanding of conceptual relationships, especially in science. Good (2011) suggested that digital game construction is an excellent way to learn the computational thinking skills that Wing (2006; 2009) suggests are fundamental to twenty-first century problem solving.

Educating pre-service teachers in the use of digital games to support teaching and learning is something fairly new for most teacher education programs (Franklin & Annetta, 2011). The challenge of preparing teachers to incorporate digital games to support pedagogical practice may also be impacted by their previous experience of playing digital games. Hayes and Ohrnberger, (2013) recently reported that pre-service teachers "were not highly engaged with gaming or with the social and technical practices associated with gaming" (p. 172); of those who were gamers, males indicated they more engaged in playing and modifying games than their female colleagues. There are also few opportunities for pre-service teachers to gain the experience they need to effectively integrate video games into their classroom practice (Schrader, Archambault, & Oh-Young, 2011). Therefore, evidence suggests that for pre-service teaches: a) the amount and sophistication of their experience may relate to differences in gender, and b) few are properly prepared to use games to support and enhance instruction when they enter professional practice. We could find only sparse research on pre-service teachers as designers and builders of their own video games, as a precursor to increasing their understanding how these games might be effectively integrated into pedagogical practice.

It is important for pre-service teachers to have a solid skill set and knowledge base in the technology they are using to effectively integrate this technology into their future pedagogical practice (Mishra & Koehler, 2006; Mumtaz, 2000; Pierson, 2001; Williams, Foulger, & Wetzel, 2009). In this study we provided pre-service teachers with direct game-building experiences as opposed to their more familiar role as game players. The pre-service teachers were asked to create a video game of their choice as part of a pre-service educational technology course. The participants in this study had no previous game creation experience. The type of game genre they chose to create was left completely open, but they were encouraged to think about how their potential students could better learn and understand subject area content (e.g. social studies, math, science, etc.) through the process of game design and game construction. Given that the pre-service teachers in our study were not constrained by the type of game they were asked to construct, we were interested in exploring their personal genre (game type) construction preferences. We examined these game genre creation preferences with respect to gender and their relationship to the previous time they spent playing games and using social media

The next sections of the paper examine the general role of digital games to support and enhance teaching and student learning. This is followed by a focused analysis of how student digital game construction might impact classroom practice for teachers and students. We then briefly examine digital game, gender and genre differences in game play, and the possible influences they might have on the educational use of digital games in classrooms. Finally, we describe the structure of our study, the findings, and a discussion of these findings.

Digital Games in Education

Support for the popular rhetoric that digital games comprise a major source of personal entertainment, was recently released by the Entertainment Software Association (2014), when they reported that 59% of Americans play video games and the average age of gamers today is 31 years old. Reports like this support the assumption that digital games are increasingly and culturally integrated into our entertainment lives, similar to movies and TV. Like movies and TV, commercial-off-the-shelf (COTS) games may support and enhance education (Gee, 2013; Mayer, 2014; Whitton, 2014). In some cases COTS digital games are designed as educational games but these often lack a theoretical grounding in educational learning theories (Kebritchi & Hirumi, 2008). With this in mind, the line between games designed for only entertainment and those specifically designed for education is beginning to blur (Games for Entertainment and Learning Lab, 2014; Gee, 2012; 2013).

Whether digital games were originally designed specifically for education or for entertainment, makes defining their role and purpose in the classroom a very open and complex question (Tobias & Fletcher, 2011). Well-structured arguments have been presented in support of using digital games for learning such as Gee's (2013) set of learning principles in digital games. Although these academic arguments for digital game use to support teaching and learning are very compelling, the evidence for the effectiveness of learning from digital games is variable (Whitton, 2014) and this evidence often indicates negative outcomes about learning from games (Clark, Yates, Early, & Moulton, 2011). Even in the case where digital games are specifically designed to support learning Young et al. (2012) concluded, "there is limited evidence to suggest how educational games can be used to solve the problems inherent in the structure of traditional K-12 schooling and academia" (p. 62). Recently released books by Whitton (2014) and Mayer (2014) have helped to synthesize the theoretical constructs that underlie their use as a potential educational tool to support pedagogical practice and propose methods for researching the effectiveness of digital games.

Whitton (2014) identified eight different relationships between learning and games:

1. Learning *with* entertainment games: existing COTS games are used to support classroom teaching.
2. Learning *with* educational games: educational games designed with a specific instructional purpose such as teaching math skills, scientific concepts, or understanding global conflicts.
3. Learning *inspired* by games: using backgammon or poker as a basis to design algorithms and research areas of artificial intelligence.
4. Learning *within* games: the informal learning that occurs during game play, such are hand-to-eye dexterity.
5. Learning *about* games: understand the role games social, psychological, and cultural development.
6. Learning *from* games: understanding how game designers created interactions that motivate learning/playing the game and applying those same strategies to other learning contexts.
7. **Learning *through* game creation: the learning that occurs when students design, develop and implement their own game, i.e. the learning that happens through game construction.**
8. Learning *within* communities: the supportive and collaborative problem-solving skills that are developed when communities of gamers are playing together (p. 4).

Whitton's eight relationships between games and learning provide a valuable framework in which to situate educational research. In the context of this paper we focus on point seven (in bold), learning through game creation and collect evidence on the types of games pre-service teachers create.

Pre-service Teachers

The Joan Ganz Cooney Center (2014) reported that 55% of teachers indicated using video games in their class at least once a week, and of those 51% said the games were used in groups of two or more. Very few teachers reported receiving any instruction during their pre-service teacher education programs on how to use video games in teaching. Instead most learned about video games from colleagues, self-education, seminars, and online discussions (Millstone, 2012).

In an earlier informal study of pre-service teachers Schrader, Zheng, and Young (2006) found that 76% of pre-service teachers had played games and of those 83% played at least once a week (the majority of those for less than one hour), but about a fifth indicated they played three hours or more per week. More recently Hayes and Ohrnberger (2013) investigated whether the gaming practices of first year pre-service teachers' influences their attitudes towards the use of technology in their teaching. Interestingly, of the 223 pre-service teachers surveyed, 94.6% indicated they had previously played digital games, but only 93 (41.7%) indicated that they currently played video games. Females represented 86.3% of the 223 participants, and of that, 38% indicated they currently played videos games, whereas 76% of the males who responded indicated they still engage in game play. Generally, among the pre-service teachers, males reported a higher level of engagement and were more likely to describe gaming as a social activity. Two important conclusions from their work were that: a) gaming among pre-service teachers was not as prevalent as one might have expected, and b) those who were more familiar with game-related communities tended to be consumers of game content rather than producers of game content.

Kenny and McDaniel (2011) studied 58 pre-service teachers and found that only 42% played video games on a regular basis. They also reported that attitudes towards game playing were more positive after the pre-service teachers had engaged in game playing. Even those who initially had indicated a negative attitude toward games changed significantly in the positive direction after they had been engaged in a game playing activity. They suggested that pre-service teacher education "programs will need to include additional courses in the theoretical underpinnings of game play, courses on how to evaluate and integrate game technologies, and more on the types of learning that they can expect as a result of their students playing games in the their classrooms" (p. 2009).

Schrader, Archambault, Oh-Young (2011) studied 13 pre-service teachers in an immersive massively multiplayer online game (MMOG) environment to investigate how additional instructional materials that highlight benefits of MMOGs influence their perceptions of MMOGs in education. The research design situated pre-service teachers in the authentic collaborative MMOG environment where they actively participated/interacted with each other to solve problems. The results of this work indicated that teachers immersed in the game environment improved their perceptions of MMOGs as a potential educational tool. To fully appreciate the pedagogical potential of digital games, Schrader et al. (2011) suggested that pre-service teachers need to be placed in a "learning in technology" (p. 261) context, where they are immersed in an authentic game design and construction situations. They conclude, "Teacher preparation programs need to use gaming/simulation within courses and scaffold the experience for future teachers to establish how meaningful learning can occur" (p.275).

Kafai et al. (1998) used game design as an interactive learning environment for fostering students' and teachers' mathematical inquiry and studied 16 pre-service teachers engaged in educational paper-based game design activity for the purpose of mathematical instruction (teaching fractions). As a result of the game design activity, thinking patterns of pre-service teachers shifted from the 'game idea' and 'content' being separate (they called extrinsic game design), to one in which the two concepts were integrated (they called this intrinsic game design). This shift from extrinsic to intrinsic game design was facilitated by the introduction of conceptual design tools such as paper screens to draw on and directives that suggested relevant content material. The pre-service teachers used their previous knowledge of development of children's mathematical thinking to help structure their game.

In summary, although pre-service teachers are likely to have some degree of digital game playing experience, it is unlikely they have game building experience. This is understandable given that digital game construction has traditionally been taught in computer science departments and usually requires extensive knowledge of software development environments, graphics libraries, and complex languages such as C++ or Python (Carbonaro et al., 2008).

Game Construction as Tool for Learning

> … Encourage children to become game designers themselves. This requires more technological infrastructure and more support from knowledgeable people… when they get the support and have access to suitable software systems, children's enthusiasm for playing games easily gives rise to an enthusiasm for making them, and this in turn leads to more sophisticated thinking about all aspects of games… the games they can make generally lack polish and the complexity of those made by professionals designers. But the idea that children should draw, write stories and play music is not contradicted by the fact that their work is not of professional quality. I predict that within a decade, making a computer game will be as much a part of children's culture as any of these art forms (Papert, 1998, p. 3).

More than thirty years ago, Seymour Papert (1980) developed a simple but effective programming language called Logo that allowed very young children to easily write programs that demonstrated sophisticated conceptual relationships. Essentially, the Logo language enabled children to *construct* a computer program and to understand the relationship between a sequence of instructions and the output it would produce, thus allowing programming to be more accessible. Papert (1980) encapsulated his ideas on this learning in a paradigm he called *constructionism* where learners built actual artifacts (products), thereby distinguishing it slightly from the more general Piagetian notion of *constructivism* where learners build conceptual understanding (Piaget, 1970). A constructionist environment involves the actual building of a physical or virtual product while in the process of learning. The product is a vehicle to ground conceptual learning (Penner, 2000). However, as Fischer and Immordino-Yang (2002) point out, the product is less important than the actual learning process and the cognitive skill development that occurs. In this sense the product, the learning process, and cognitive skill development are innately linked.

Penner (2000) presented a strong argument for the use of model building as an important component for learning and understanding scientific concepts. Penner (2000) outlined three important ideas on why the construction of exploratory models should be an essential part of the scientific curriculum: a) building models of a phenomena help students deepen their understanding of the natural world; b) by constructing model representations of a phenomena students can dialog about their models in ways that differ predefined models they are given in schools, e.g., scientific theories; and c) model construction affords the student the opportunity to reflect their thinking during and after the construction process.

Penner's (2000) arguments for the importance of students' constructing scientific models are analogous to arguments for students' construction of digital games. First, in the case of building a game, students have the ability to deepen their conceptual understanding for both a content area and the computational thinking process that is required to produce a video game. Students can build digital games that incorporate scientific concepts (collision detection) or social justice concepts (Native American fur trade in the 1800s) or both within the same game (Salerno, 2013). Klopfer et al. (2009) suggested that students and teachers building simulations (scientific models) in the form of games, experience algorithmic and design thinking and the ability to use programming itself as a curriculum tool.

Second, COTS games for either entertainment or education are analogous to what Penner (2000) called 'finished' representations that often typify science education and as such, are biased by the choices made by the game designers. In direct contrast, as the designer of their own game, a student can discuss their product with respect to the choices they make, and how they incorporate elements into the design and building process. Kafai (1996) reported that the design process was more important than the actual game (artifact) produced by the students and that students often went beyond the thinking and learning they did in a traditional classroom.

Finally, students constructing their own games have the opportunity to discuss the building process and their final game product analogous to the way Penner (2000) suggested student's would discuss the scientific models they created; "Once reified, students' models are open to inspection, evaluation, personal reflection, and public discussion" (p. 28). Student reflection on their own learning is an important component toward deepening their personal understanding (Baird et al., 1991). Gee (2013) also emphasizes the important role that reflection has in video game play for learning and the sharing of ideas.

Unfortunately, the ability to construct video games with any level of sophistication has, until recently, been severely limited by the difficulty required to master complex computer programming languages and their supporting toolsets. The result is that a student's ability to create, express and construct ideas through the building of a digital game is often regulated to a paper based design and cannot be realized unless it is handed over to a programmer. During the past several years, and building from Papert's early ideas, new game development toolsets and simplified but powerful programming languages have become available. They allow digital game creation to move out of the hands of experienced programmers and into the hands of children and adolescents (Carbonaro, Szafron, Cutumisu, & Schaeffer, 2010). For example, *ScriptEase* (BELIEVE, 2014), *Storytelling Alice* (Kelleher, 2007), *Scratch* (MIT, 2009), and *Kodu*

(Microsoft, 2014) represent just some of these new, easily accessible, video game development environments.

Good (2011) identified a number of these programming environments that are often freely available for novice programmers (e.g. *Scratch*, *Alice*, *Looking Glass*, *Greenfoot* and *Flip*). Children and young people are using these environments in creative ways to construct video games, simulations and interactive stories. Good (2011) further suggests using new programming environments can help promote the development of computational thinking skills that are increasingly important for problem solving in today's world (Grover & Pea, 2013; Riley & Hunt, 2014; Wing, 2006). For example, Salen (2007) described her work on the development of *Gamestar Mechanic* as a tool that allows players to design their own Role Playing Game and by doing so learn the fundamentals of game design. Observation of grade six to eight students using the *Gamestar* tool indicated that when given a choice to modify an existing game or create something new, most choose to create their own game. She argues that game design can teach system thinking and design skills that are important computational thinking constructs.

Carbonaro et al. (2008) demonstrated that the *ScriptEase* tool could augment a commercial off-the-self game software development toolset to enable grade ten students to construct digital games. Results from this research showed that students could construct video games with limited instruction and that gender was not a significant factor in the quality of these games, regardless of previous programming experience and/or computer game playing experience. The authors' conclude by saying, "in the educational context of this study, *ScriptEase* provides an easy-to-use tool for interactive story authoring in a constructionist learning environment." (p. 687).

Vos et al. (2011) compared elementary level students on the differences between motivation and deep strategy (i.e., critical analysis of ideas, linking concepts and principals, and problem-solving) with respect to game playing versus game construction. The results of their work demonstrated significant differences in favour of increased intrinsic motivation and deep strategy used for the game construction condition.

Robertson and Howells (2008) used computer game design and construction as ways to promote student learning and identified that the teacher role needs change from that of the provider of information to that of a facilitator of information. During the game design process teachers “will need to judge carefully the moment at which to intervene with focussed instruction so that the creative flow is not lost, and embrace the opportunity for pupils to teach each other” (p. 577).

Boechler, Artym, DeJong, Carbonaro, & Stroulia (2014) analyzed games created by pre-service teachers and found that those who played more hours of digital games actually used a higher number of variables in their games, resulting in more complex games. They suggest that the more exposure pre-service teachers have to playing digital games, the more these teachers may grasp the mechanics that underlay digital games.

Project schools are beginning to appear that have a significant focus on students using, designing and constructing video games. For example, Quest to Learn (q2l.org), for grades 6 to 12 in New York City, uses an inquiry-based approach, which includes the design of complex problem spaces to help students self-evaluate their understanding of content material and their personal problem-solving (Salen, 2011). For teachers to use digital games in these and similar types of instructional environments, it is important they know how to effectively integrate game building programming projects into the classroom. For example, children in grade six and seven might use the *Kodu* (Microsoft, 2014) programming environment to build either a game on the fur trade in the early 1800s, or one on recycling that teaches them about aspects of agriculture (Boechler, Carbonaro, deJong, Stroulia & Gutiérrez, 2011). Various courses and resource material (e.g., code.org, codeacademy.com, learnscratch.org) are appearing to help support teachers' with programming and using programming environments. For example, The Irish Computer Society offers a fully funded Teacher Training (primary or secondary) course on the use of the *Scratch* programming environment (ICS, 2014).

Gender and Genre in Digital Gaming

As digital game playing and construction makes its way into teacher education programs and subsequent professional practice, there are still concerns around the genre of game students are building and its relations to issues of gender (Robertson, 2012). Almost all children/teens (12-17 age) report (97%) playing video games at some point; however, boys report playing more often and for longer periods than girls (Lenhart, et al., 2008b). Although the amount and duration of game play drops off somewhat in the later teen years due to other commitments, overall game play accumulates such that males often far exceed females in time spent gaming (Heeter et al., 2008). Furthermore, Lenhart et al., (2008b) reported that boys played an average of eight different genres whereas girls reported playing an average of six. More boys reported playing first-person shooter fighting games whereas girls reported more puzzle-type games. Lenhart et al. (2008b), also found that 50% of the boys indicated they currently play games with an M (Mature) or AO (Adult Only) rating as one of their top three games in comparison to only 14% of girls who reported they played these types of games.

Hayes (2011) states "the significance of gender in the context of educational use of video gaming remains a contested and potentially confusing topic" (p. 453). Hayes outlines the challenges of understanding relationship between gender, gaming and learning in the context of sociocultural values and norms. In this sense there is a need to consider many factors and variable relationships such as the cultural context, game genres, variability in game play, and social interactions in gaming, to name a few. The space of these relationships is truly multidimensional; therefore, gender itself cannot be the single defining factor for preferences in the game genre players select to play. Hayes argues that the varied purposes and approaches to using games for learning need to be considered when evaluating the significance of gender. Consequently, given that there are significant differences in total gameplay time between males and females, Jenkins and Cassells (2009) argue that it would be mistake to introduce digital games in the classroom, if by doing so in some way, it disadvantages females.

Bourgonjon, Valcke, Soetaert, Schellens (2010) studied 858 students ages 12 – 20, composed of 48.1% female and 51.9% males. The researchers evaluated a set of variable relationships: gender, preference for video games, usefulness, ease of use, and experience. They found significant gender differences in game playing hours per week (after adjusting for statistical anomalies removing top and bottom 5% outliers). Males reported significantly more playtime hours per week (6.96) whereas females reported only (2.16). Although males reported being significantly more in favour of using video games in schools, there were no gender differences on the belief that video games offered learning opportunities. The researchers proposed a path model that indicates experience with game playing, perceived learning opportunities of games, and usefulness of games in schools as all have a mediating role in student preference for video games in schools. As a result, the researchers argue that the teacher should explain the advantages of using games as a tool for learning and have students design their own games.

Lenhart, Jones, and Macgill (2008a) in their survey of adult video gaming reported gender differences in game modification (modding), with 36% of males reporting they modified games as opposed to 20% of the females. Hayes (2008) surveyed 1139 children in grades five to nine on game content creation and information technology proficiency. In-game content creation comprised of making additional levels, new characters or objects, add-ons, and cheat codes. The results indicated that boys were significantly more likely than girls to create in-game content. The most popular in-game content creation for the boys (35.7%) was cheat codes whereas for girls it was new characters, clothes, and other items (20.1%). Hayes suggested that gender differences for in-game creation may be related to the types of games girls play and that these games often do not have in-game creation capabilities. Conversely, boys are more heavily involved in the online gaming communities that promote, share, and value in-game creation contributions.

Although there is limited research on the relationship between gender and game construction, what little there is often shows either no differences on gender or positive differences in favour of females for learning specific skills such as those associated with storytelling and/or computing science. Denner et al. (2011) provides evidence that middle school girls can use game construction as a way to learn computer science concepts. Similarly, Robertson (2012) studied game creation by children (11-12 years old) and reported that girls scored higher than boys on their skills related to storytelling. Carbonaro et al. (2010) showed that digital game construction of RPGs (stories) by students (14-15 years old) was an enjoyable process that promotes the teaching of higher-order thinking skills and the learning of computing science concepts. The outcome measures were gender-neutral and they argued that video game development (as opposed to video game playing) could be a possible factor to motivate females to pursue computing science as an area of study. Vos et al.'s (2011) survey on students (10-12 years old) also reported similar results with respect to positive outcomes for learning and enjoyment on their questions of game construction; no gender differences were reported.

Methods

Participants

The pre-service K-12 teachers involved in this study chose EDIT 486, the course outlined in this paper, as an elective at a large Western Canadian university. Students had the choice whether or not to provide consent to participate in this study. A large majority of the students provided consent. In 15 course sections over a 48 month period, we had 166 participants: 60 (36%) females and 106 (64%) males. While this sample size was relatively small, the group is representative of diverse demographics and academic focuses within the field of Education.

Context

EDIT 486 is a senior undergrad, educational technology elective course taught at the University of Alberta through the Educational Psychology department. The first iteration of the course was outlined by Boechler et al. (2011). In this course, pre-service teachers become builders of video games showcasing constructionism in practice. By the end of the course, pre-service teachers should be able to use video games in their classrooms and implement the acquired pedagogy. Students use the graphic programming environments *Scratch* (MIT, 2009) and *Kodu* (Microsoft, 2014), utilizing one environment for the first half of the semester and the other for the second half.

The course has a 50/50 lecture/lab format running twice a week in three hour periods. The lab period is further divided into two components: for the first half a teaching assistant, who is well-versed in *Scratch* (MIT, 2009) and *Kodu* (Microsoft, 2014), walks students through essential techniques for game design. The second half of this time period is 'open lab' time for students work on assigned tutorials derived from the textbook *Scratch for Teens* (Ford, 2008). Students are required to complete three of these short tutorials a week where they learn the basics of the interface. In addition, students are encouraged to personalize the assignments to better suit their understanding of the interface, as well as to explore potential ideas for their *Scratch* (MIT, 2009) video game project. For the duration of the 'open lab' time the teaching assistant is on hand to answer any questions students may have.

After students complete all the tutorials, they must make a *Scratch* (MIT, 2009) video game project on a topic of their choice. These *Scratch* (MIT, 2009) projects are examined in this paper. As the course focuses on constructionism, students are encouraged to build their knowledge of a topic into the game as opposed to directly making an educational game. For example, a student's understanding of gravity could be developed into a game where gravity is constantly acting on a game character as opposed to using a quiz or kill and drill game expressing the physics formulas used to represent the force of gravity. Students receive two dedicated weeks to complete their projects after their tutorials are complete, but can start working on their projects at the start of the course. After completing the project, students write a personal reflection on the game design process concentrating on their areas of learning from their experience creating the game.

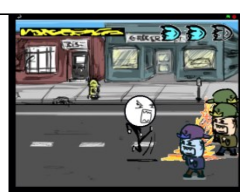
The lecture covers various types of constructivism, constructionism, technology taxonomies, video game literacies, integrating video games into the classroom, programming basics, and applications in education, among others. Students have weekly assigned readings that they take turns presenting on in groups. These presentations include an online discussion, an in-class discussion as well as an activity that utilizes a technology to go over the main points of the readings. The final paper for this course allows the students to create a plan on how they would implement *Scratch* (MIT, 2009) or *Kodu* (Microsoft, 2014) in a classroom in their choice of subject area. They must create a lesson plan and evaluation scheme and support their decisions with citations from the literature.

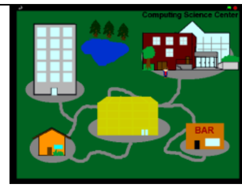
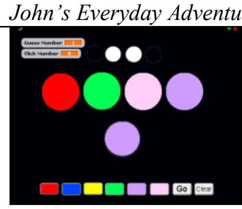
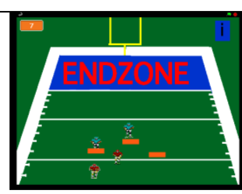
Construction of the Genre Classification Scheme

A classification scheme was constructed to evaluate the games based on genre. No one schema has been universally agreed upon. Wolf (2008) identified over forty different video game genres, which communicates the complexity of clarifying genres. “The idea of genre has not been without difficulties, such as the defining of what exactly constitutes a genre, overlaps between genres, and the fact that genres are always in flux as long as new works are being produced (p. 1). Henry (2011) wrote his dissertation on the evolution of genre and put forward that most authors and publications use some form of the genres: Action, Adventure, Driving/Racing, Puzzle, Simulation, Sports and Strategy. Table 1.1 shows the genres this study identifies through a comprehensive literature review as well as being present in our dataset (Adams, 2007; AllGame, 2010; Apperley, 2006; Elliott, Ream, McGinsky, & Dunlap, 2012; ESA, 2013; GameFAQs, 2014; GameFly, Inc., 2003, February 28; Hamlen, 2011; Henry, 2011; IGN, 2014; Lee, Ko, Song, Kwon, Lee, Nam & Jung, 2007; Lucas & Sherry, 2004; Rollings & Adams, 2003; Ventura, Shute & Jeon, 2012; Whitton, 2014;Wolf, 2008).

Table 1.1

Genre classification scheme in our study with examples, findings and sources.

Genre	Summary	Game Example	Observed Qualities
Action	This type of game focuses on reactions and timing.		
Action - Violent	Focus on violence as a major game mechanic.	 *Felix Fury and the Frosting Phantom*	● Score points ● Attempt to get high score ● Reaction tests ● Space and zombie themes

Action - Non-Violent	Action game with non-violent motif.	 *Cow Ski Valley*	• Commonly collecting objects • Commonly avoiding other characters or objects • Quick-time events requiring quick reactions • Getting points • Trying to achieve a high score • Platformers • Frogger-type games • Pong and Brickbreaker-type games
Adventure	This type of game focuses on player interactions based on developing the narrative.	 *John's Everyday Adventure*	• Narratives • Fantasy-themed games - dragons, kings, swords and princess
Puzzle	This type of game focuses on logical reasoning and problem solving.	 *Think Tank*	
Artificial Life	Modeling biological processes.	 *Gecko Tank*	• Animals - notably reptiles (turtle and gecko)
(Construction And Management) Simulation	This type of game focuses on building a structure while managing resources.	Students in our class have not used *Scratch (MIT, 2009)* to create a Construction and Management Simulation type game.	
Role Playing	This type of game focuses on narratives emphasizing character development.	Students in our class have not used *Scratch (MIT, 2009)* to create a Role Playing type game.	
Sports	This type of game focuses on athletic competition.	 *All Pro Football*	Only two Sports games were made.
Strategy	Resource management-style game play.	 *Monster Dungeon*	Only one strategy game was made.

Driving/Racing/ Vehicle Simulation	This type of game focuses on operating vehicles.	 Unnamed driving game.	Only one Driving/Racing/Vehicle Simulation game was made.
Other		 Unnamed educational game	• Some educational games - label body parts or geographical areas • Song creation soundboards

GameFAQs (2014) and IGN (2014) are two gaming website communities commonly used by gamers. In 1995, GameFAQs was founded and primarily contained cheat codes, frequently asked questions and walkthroughs for video games. IGN, the Imagine Games Network, was created in 1996 and primarily reviews commercial games before they are released. Both websites were also consulted in defining our schema.

Action. Action games are driven by the manipulation of onscreen elements (Allgame, 2010). Elements common to action games include: lives, quick-time events (reaction tests) and hand-eye coordination tests; often these games include levels, waves and power-up design elements (Rollings & Adams, 2003). Goals may be easy to perceive, but difficult to accomplish (Adams, 2007). Action games commonly also have checkpoints or enemy spawn points (Adams, 2007). Action games only display the information players need to know and whenever possible, this is done through graphical indicators instead of numbers or text (Adams, 2007).

Rollings and Adams (2003) differentiate action games into two categories: those with shooting (Action Shooter) and those without (Action Non-Shooter). This ‘shooter’ distinction can be somewhat misleading as the categories more so refer to the violent content in the games. Games containing violence, usually with weapons, fit into the Action Shooter category, whereas non-shooter action games have a non-violent theme. We have relabeled these categories to reflect the more inclusive definition that Rollings and Adams (2003) applied: Action Violent and Action Non-Violent.

Action violent. Violent action games tend to have two main characteristics: hit the enemy and avoid being hit. For example, in a Shooter game, the player must shoot the enemies, while refraining from being shot (Allgame, 2010). This type of game play has been called "kill-or-be-killed" (Elliott, Ream, McGinsky, & Dunlap, 2012, p. 954). Subgenres of video games that would fit in this category include: Shooters, Fighting and Violent Platformers. Commercially Off-the-Shelf (COTS) available examples of this type of game include:

- *Call of Duty: Advanced Warfare*
- *Far Cry 4*
- *Assassin's Creed Unity*

Action non-violent. These are action-based games requiring quick reactions or coordination, but do not emphasize combat of any form. A non-violent platform game would be one that involves dodging collisions with enemies while traversing an environment requiring precision timing of movement and jumping to reach an objective (Elliott, Ream, McGinsky, & Dunlap, 2012). COTS available examples of this type of game include:

- *LittleBigPlanet 3*
- *Donkey Kong*
- *Super Monkey Ball 2*

Adventure. Rollings and Adams (2003) identify an Adventure game as a game centered on a character immersed in an interactive story. Some Adventure games can take over 40 hours to complete the first time, but if the player knows how to solve the puzzles and what routes to take, this could be reduced to 4 hours (Adams, 2007); Wolf (2008) elaborates that these type of games are non-linear, and thereby allow the player to explore the game's environment. Action and other elements may be present in an Adventure game, but these elements are secondary to exploration and advancement of the narrative (Allgame, 2010). Common conceptual puzzles in these games include: finding keys to locked doors, obtaining inaccessible objects and collecting things (Adams, 2007). COTS available examples of this type of game include:

- *The Walking Dead*
- *Lego Batman 3*

Puzzle. Rollings and Adams (2003) define puzzle games as ones that primarily center on the act of puzzle solving; sometimes these puzzles are separate from a narrative. COTS successful puzzle games need to be challenging, visually attractive and enjoyable (Adams, 2007). Games usually involve matching and logical reasoning (Elliott, Ream, McGinsky, & Dunlap, 2012). COTS available examples of this type of game include:

- *Scribblenauts Unmasked: A DC Comics Adventure*
- *Pokémon Battle Trozei*
- *Tetris*

Artificial life. Artificial Life games model biological processes (Rollings & Adams, 2003). These digital creatures can die without proper care from the player (Wolf, 2008). COTS available examples of this type of game include:

- *Tamagotchi*
- *Spore*
- *The Sims*

Construction and management simulation. Construction and Management Simulation games are about balancing limited resources to build or expand a business, community or empire while contending with other internal variables (i.e. Crime; Wolf, 2008); the player can control building locations, factors to reduce prices, and also demolition (Rollings & Adams, 2003). There is an emphasis on cause-and-effect (Allgame, 2010). COTS available examples of this type of game include:

- *Roller Coaster Tycoon*
- *SimCity*
- *FTL: Faster Than Light*

Role Playing. Role Playing games also contain strong storylines, but focus on player characters that improve through experience over the duration of the game. The in-depth storyline allows the player to sympathize with the main characters (Rollings & Adams, 2003). Commonly the main character completes minor quests or tasks contributing to the larger, central goal in the game (Allgame, 2010). Role playing games tend to revolve on the concept of the creation of a customized, powerful party (Elliott, Ream, McGinsky, & Dunlap, 2012). COTS available examples of this type of game include:

- *Diablo III*
- *Final Fantasy X*
- *South Park: The Stick of Truth*

Sports. Sports-type games are based on athletic competition (Rollings & Adams, 2003). COTS available examples of this type of game include:

- *Mario Golf World Tour*
- *FIFA 15*
- *Tony Hawk's Pro Skater HD*

Strategy. Strategy games focus on resource management in either a turn-based or real-time environment. Commonly they have three primary factors: the theme, the presentation layer and multiple perspectives (Rollings & Adams, 2003). This style of game play emphasizes strategy instead of fast action and quick reflexes (Wolf, 2008). Player decisions have cascading, long-term effects (Allgame, 2010). COTS available examples of this type of game include:

- *Sid Meier's Civilization V*
- *Total War: Rome II*
- *Command & Conquer*

Driving/racing/vehicle simulation. Rollings and Adams (2003) identify a game as a Vehicle Simulation game if the player is immersed in a vivid driving/flying situation, real or imaginary; these vehicles can include: flight, driving, boats or ships, tanks or spacecrafts. Driving/Vehicle Simulation-type games focus more on creating a genuine driving experience, whereas this may be secondary in a Racing-type game as the primary goal is to get from one area to the finish line as quickly as possible. Wolf (2008) writes that Driving games focus on steering, manoeuvrability, speed control, and fuel conservation. Overall these genres are about operating a vehicle. COTS available examples of this type of game include:

- *Mario Kart 8*
- *Gran Turismo 6*
- *Blazing Angels*

Data Collection

In order to anonymize the students' data, unique identification numbers were randomly assigned to the students. All their game data, demographic data is tied to their ID number, with no biased pattern linking them to their gender, the term that they took the course, or any other data collected. Two instructors jointly taught all the sections the course.

Video Game and Social Media Experience

Students were asked to complete the Computer Experience Questionnaire (Boechler, Leenaars, & Levner, 2008; see Appendix A) which measures the frequency of recreational use of computers including the duration and intensity of experience with applications such as video games and social media platforms. The questionnaire asks students to report how many hours a week they spent playing video games and how many hours a week they spent on social media sites, with reports from none at all to over 10 hours a week.

Genre Classification

Using the genre classification scheme (Table 1.1), two trained experts both with experience playing video games as well as teaching in environments using video games (the EDIT 486 course as well as Department of Computing Science Summer Camps on building video games), classified all the games; interrater reliability for a randomly selected 29 percent (N = 49) of the projects (166 total projects) was .9 (Cohen's Kappa). While some of these games may have more than one genre, they were categorized according to what the games' primary components are.

Game genres are not necessarily exclusive. Games that fall in one genre may fall under another. All games were assigned primary genres, secondary and so on. Few games had secondary genres and, in addition, any analysis of secondary genres would not be independent of the primary genre. Thus, these classifications were omitted. As an example, Grand Theft Auto V is an Action Violent game, but has elements of Adventure as it has a strong narrative and many racing components which would also lend it to the Driving/Racing/Vehicle Simulation genre.

The defining characteristics that place it in the Action Violent category would be the fact that its overall game concept revolves around combat and transport from point A to B.

Data Analysis

All analyses were performed by SPSS 21.0 software (SPSS Inc., USA). For all models and variables, there were no indications of multicollinearity as tolerances were well above the suggested minimum of .2. As the data is categorical, Chi-square was the primary analysis used.

Gender and Weekly Average Hours of Video Game Playtime

First, the results of the Computer Experience Questionnaire, focusing on video game hourly usage per week, were compared to the gender data to see if there was any association. Tables 1.2 through 1.5 summarize the frequencies.

Table 1.2
Frequency table showing the number of hours of video games played per week in recent years, by gender (self-reported).

	Range of video game hours played per week					Total	*M*	*SD*
	None at all (1)	1-3 Hrs (2)	4-6 Hrs (3)	7-10 Hrs (4)	>10 Hrs (5)			
Female	25	11	8	3	1	48	1.833	1.0586
Male	15	29	17	9	13	83	2.711	1.3208
Total	40	40	25	12	14	131	2.389	1.2984

Table 1.3
Frequency table showing the number of hours of video games played per week in high school, by gender (self-reported).

	Range of video game hours played per week					Total	*M*	*SD*
	None at all (1)	1-3 Hrs (2)	4-6 Hrs (3)	7-10 Hrs (4)	>10 Hrs (5)			
Female	20	18	2	5	3	48	2.021	1.2115
Male	9	16	14	16	28	83	3.458	1.4082
Total	29	34	16	21	31	131	2.931	1.504817

Table 1.4
Frequency table showing the number of hours of video games played per week in junior high school, by gender (self-reported).

	Range of video game hours played per week					Total	*M*	*SD*
	None at all (1)	1-3 Hrs (2)	4-6 Hrs (3)	7-10 Hrs (4)	>10 Hrs (5)			
Female	17	19	6	4	2	48	2.063	1.0994
Male	9	13	20	21	20	83	3.361	1.3027
Total	26	32	26	25	22	131	2.885	1.3792

Table 1.5
Frequency table showing the number of hours of video games played per week in elementary, by gender (self-reported).

	Range of video game hours played per week					Total	***M***	***SD***
	None at all (1)	1-3 Hrs (2)	4-6 Hrs (3)	7-10 Hrs (4)	>10 Hrs (5)			
Female	20	22	1	2	3	48	1.875	1.0842
Male	15	20	23	8	17	83	2.904	1.3759
Total	35	42	24	10	20	131	2.527	1.3662

Our first hypothesis was that there is no association between average numbers of hours playing video games in any time period (e.g., elementary, junior high, high school, university or overall) and gender. Chi-square analyses were used to explore possible relationships between gender and the previous average numbers of hours playing video games. Chi-square analysis revealed a statistically significant interaction between the amount of hours over various time periods and gender: gender and game play hours reported over post-secondary years ($\chi^2_{(4)}$ = 19.141, p = 0.001), gender and game play hours reported over high school years ($\chi^2_{(4)}$ = 32.158, p = 0.000), gender and game play hours reported over junior high school years ($\chi^2_{(4)}$ = 26.949, p = 0.000), gender and game play hours reported over elementary years ($\chi^2_{(4)}$ = 20.435, p = 0.000) and gender and game play hours reported taken as a sum ($\chi^2_{(16)}$ = 40.153, p = 0.001). Bonferroni's correction supported the alpha values are still found to be significant.

In all the time periods explored, males played more hours of video games as oppose to their female counterparts. The relationship between these two variables is also quadratic, which is shown in Figure 1.1. Students play fewer hours of video games in elementary and university, and more in junior high and high school, regardless of gender.

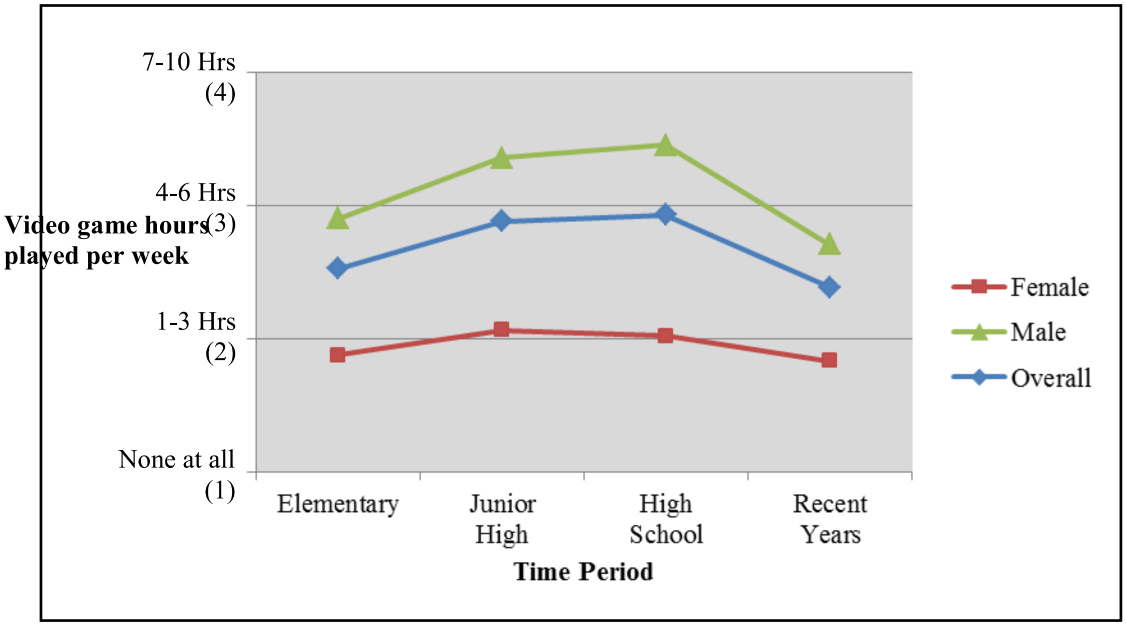


Figure 1.1. Average range of hours spent playing video games over periods of time. This figure illustrates different genders' mean playtime ranges.

To gain a more complete understanding of the relationship between the number of hours pre-service teachers played per week on average and the period of their life (e.g., university, high school), a multinomial linear regression was used to analyze each of the item responses. The intervals (e.g., none at all, 1-3 hours) were compared for every time period (e.g., university, high school) to see if males or females were more likely to play more hours per week, using none at all as a baseline. What follows is the results of this analysis.

Hours a week playing video games in elementary. It was found that in elementary school, males were more likely than females to play 4-6 hours of video games than none at all (Exp(B) = 1.212, p = 0.001), and >10 hours of video games than none at all (Exp(B) = 5.333, p = 0.005). Males are more likely than females to play 4-6 hours or greater than 10 hours of video games in junior high, but the 7-10 hour range is not significant at the 0.05 alpha level.

Hours a week playing video games in junior high. It was found that in junior high, males were more likely than females to play 4-6 hours of video games than none at all (Exp(B) = 6.296, p = 0.003), 7-10 hours of video games than none at all (Exp(B) = 9.917, p = 0.001) and

>10 hours of video games than none at all (Exp(B) = 18.889, p = 0.001). Males are more likely than females to play upwards of 4 hours of video games in junior high.

Hours a week playing video games in high school. It was found that in high school, males were more likely than females to play 4-6 hours of video games than none at all (Exp(B) = 15.556, p = 0.001), 7-10 hours of video games than none at all (Exp(B) = 7.111, p = 0.003) and >10 hours of video games than none at all (Exp(B) = 20.741, p = 0.000). Males are more likely than females to play upwards of 4 hours of video games in high school.

Hours a week playing video games in university. It was found that while in university, males were more likely than females to play 1-3 hours of video games than none at all (Exp(B) = 4.394, p = 0.002), 4-6 hours of video games than none at all (Exp(B) = 3.542, p = 0.019), 7-10 hours of video games than none at all (Exp(B) = 5.000, p = 0.030) and >10 hours of video games than none at all (Exp(B) = 21.667, p = 0.005). Males are more likely than females to spend time playing video games than females while in university.

Gender and Weekly Average Hours of Social Media Usage

Social media average weekly usage was compared to the gender data to see if there was a relationship to hourly usage per week. Tables 1.6 through 1.9 summarize the frequencies.

Table 1.6
Frequency table showing the number of hours of social media used per week in recent years, by gender (self-reported).

	Range of social media hours consumed per week					Total	***M***	***SD***
	None at all (1)	1-3 Hrs (2)	4-6 Hrs (3)	7-10 Hrs (4)	>10 Hrs (5)			
Female	0	16	16	8	8	48	3.167	1.0785
Male	4	27	28	10	14	83	3.036	1.1524
Total	4	43	44	18	22	131	3.084	1.1234

Table 1.7
Frequency table showing the number of hours of social media used per week in high school, by gender (self-reported).

	Range of social media hours consumed per week					Total	***M***	***SD***
	None at all (1)	1-3 Hrs (2)	4-6 Hrs (3)	7-10 Hrs (4)	>10 Hrs (5)			
Female	10	17	12	7	2	48	2.458	1.1101

Male	17	32	23	4	7	83	2.422	1.1275
Total	27	49	35	11	9	131	2.435	1.1170

Table 1.8

Frequency table showing the number of hours of social media used per week in junior high, by gender (self-reported).

	Range of social media hours consumed per week					Total	*M*	*SD*
	None at all (1)	1-3 Hrs (2)	4-6 Hrs (3)	7-10 Hrs (4)	>10 Hrs (5)			
Female	18	20	4	4	2	48	2.000	1.0916
Male	43	21	9	7	3	83	1.867	1.1345
Total	61	41	13	11	5	131	1.916	1.1166

Table 1.9

Frequency table showing the number of hours of social media used per week in elementary, by gender (self-reported).

	Range of social media hours consumed per week					Total	*M*	*SD*
	None at all (1)	1-3 Hrs (2)	4-6 Hrs (3)	7-10 Hrs (4)	>10 Hrs (5)			
Female	46	1	1	0	0	48	1.063	0.3200
Male	74	4	3	1	1	83	1.205	0.6764
Total	120	5	4	1	1	131	1.153	0.5748

There was no significant interaction between average numbers of hours spent on social media (different periods of time) and gender. Interestingly, both genders saw an increase in amount of time spent using social media as they grew older as shown in Figure 1.2.

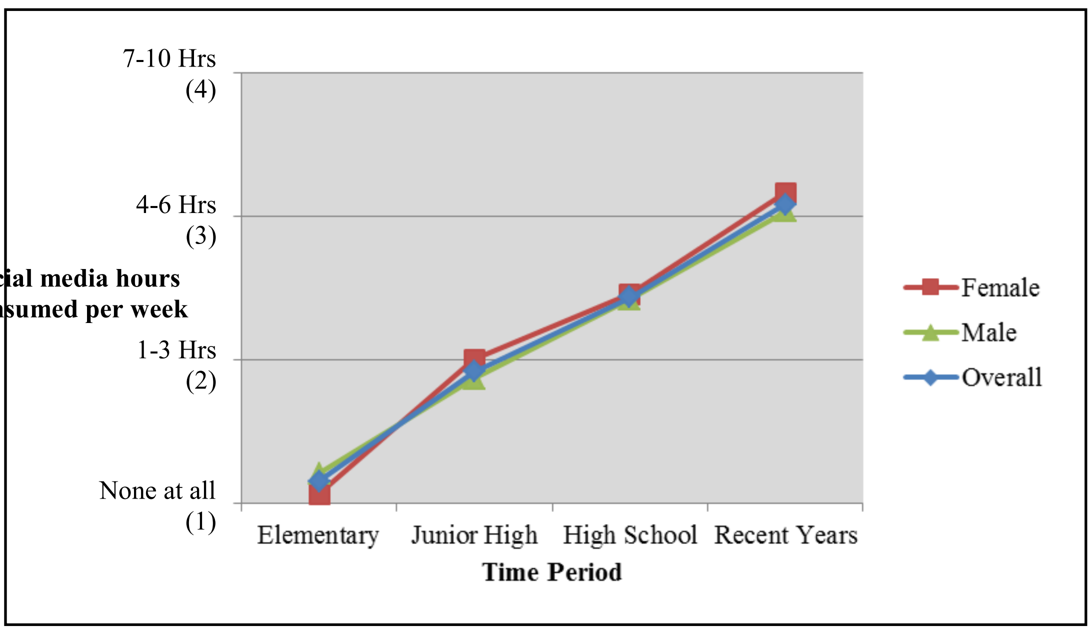


Figure 1.2. Average range of hours spent on social media over periods of time. This figure illustrates different genders' mean usage ranges.

Gender and Genre

Table 1.10 below shows the descriptive statistics for our dataset revealing which gender of students created which genre of game. Table 1.1 provides examples of each of genre and common characteristics for each genre.

Table 1.10
Video game genres constructed, by gender (observed frequencies).

	Genre									Total
	Action Non-Violent	Action Violent	Adventure	Artificial Life	Puzzle	Sports	Strategy	Driving	Other	
Female	32 (53%)	2 (3%)	6 (10%)	2 (3%)	12 (20%)	0 (0%)	0 (0%)	0 (0%)	6 (10%)	60
Male	37 (35%)	33 (31%)	14 (13%)	1 (1%)	11 (10%)	2 (2%)	1 (1%)	1 (1%)	6 (6%)	106
Total	69 (42%)	35 (21%)	20 (12%)	3 (2%)	23 (14%)	2 (1%)	1 (1%)	1 (1%)	12 (7%)	166

Our game genres were collapsed from nine genres to five different genres based on their conceptual relationships in order to meet the assumptions of the Chi-square tests. Our Action genre was singularly larger than other genres. With analysis this was further distributed into Action Violent and Action Non-Violent. The Female-Action Violent cell does not have a minimum of five samples, but this is because females tend not to create Action Violent types of games. Miscellaneous games could not be grouped into other categories and their categories were also not large enough for valid analysis. These categories include: Artificial Life, Construction and Management Simulation, Role Playing, Sports, Strategy and Driving/Racing as well as the other games (educational games, card games, music soundboards and projects that weren’t actually games: no objectives). As the data is categorical, Chi-square was the primary analysis used.

When the games in the Driving and Sports categories were examined, they were found to have non-violent motifs placing them into the Action- Non-Violent category as they also focus on timing and precision. Puzzle and Strategy games focus on logic so Strategy was combined into the category for analysis. Artificial Life was added to Miscellaneous.

Table 1.11
Video game collapsed genres, by gender (observed frequencies).

	Genre					Total
	Action Non-Violent	Action Violent	Adventure	Puzzle	Miscellaneous	
Female	32 (53%)	2 (3%)	6 (10%)	12 (20%)	8 (13%)	60
Male	40 (38%)	33 (31%)	14 (13%)	12 (11%)	7 (7%)	106
Total	72 (43%)	35 (21%)	20 (12%)	24 (14%)	15 (9%)	166

Our third hypothesis tested was that there was no association between gender of game creator and genre of game created. Chi-square analysis revealed a statistically significant interaction between gender and genre of game created ($\chi^2_{(4)} = 20.435$, $p = 0.000$). A multinomial logistical regression was performed to determine how much more likely males are than females to create a violent action game. It turns out males are more likely to make an Action Violent game than any other genre of game ($p < 0.05$) than females [Action Non-Violent (Exp(B) = 0.076, $p = 0.001$), Adventure (Exp(B) = 0.141, $p = 0.026$), Puzzle (Exp(B) = 0.061, $p = 0.001$), Miscellaneous (Exp(B) = 0.053, $p = 0.001$)]. Males are also much more likely to construct violent action games over any other type of game than females are. There were no significant results' regarding female's building preferences.

Genre and Weekly Average Hours of Social Media Usage

Our fourth hypothesis was that there is no association between average numbers of hours spent on social media (elementary, junior high, high school, university or overall) and genre. Chi-square analyses were used to explore possible relationships between the types of video games created (genre) and the creator's previous average amount of time spent on social media. There were no significant interactions between the average numbers of hours spent on social media spent and genre.

Genre and Weekly Average Hours of Video Game Playtime

Our fifth hypothesis was that there is no association between average numbers of hours playing video games (elementary, junior high, high school, university or overall) and genre of game created. Chi-square analyses were used to explore possible relationships between the average numbers of hours playing video games (at different ages) and the genre of the students' game. There was no significant interaction between average numbers of hours playing video games (different periods of time) and genre.

Summary and Discussion of the Results

With the advent of new user friendly tools for game creation, teachers can now empower their K-12 students with the possibility of learning through designing and building their own personal digital games. The game construction process can offer these students an alternative way to represent and instantiate their knowledge of a content area while supporting their learning of computational thinking skills. For pre-service teacher education programs to help effectively prepare the use of game construction in classroom practice, it important to provide pre-service teachers with a deeper understanding of how games are designed and built. A situated, hands-on constructionist learning experience is one way this can be accomplished. In this situated context where the pre-service teacher is an apprentice designer and digital game builder, thereby learning important skills. An important set of questions that guided this study: What, if any, predisposition do pre-service teachers have in a digital game creation context when they are free to choose the type of game they are allowed to create? If they are predisposed to the creation of a specific type of genre, is that predisposition related to issues of gender, previous amount of time playing games or the use of social media? Understanding a pre-service teacher's predispositions can help teacher education programs to design instructional experiences that recognize and compensate for these predispositions in the area of educational technology. Therefore, the purpose of this research was to investigate the preferences that pre-service teachers may have in digital game construction as it relates to genre, gender, and previous time spent playing digital games and with social media.

The pre-service teachers who participated in the study were enrolled in an elective Educational Technology course. An important goal of the course was to learn digital game construction skills that could be used to support and enhance their future K-12 teaching. In our study we found that male pre-service teachers preferred to create violent action games to any other type of genre. This could be a reflection of male's preference for playing violent video games (Rollings & Adams, 2003; Phan et al., 2012), or the result of a complex set of social and cultural issues that are related to gender and gaming (Hayes, 2011) which would be beyond the scope of this study. Our data did not present any discernible genre patterns for the games that females prefer to create; although, regardless of gender, 63% of these pre-service teachers preferred to create games in the action genre category. This might be due to the fact that action games are the most widely played of all games (Anderson & Bavelier, 2011). In our study we grouped the Shooter category under action based on the skills associated with game play and a complete literature review. As defined in our study, Action games comprise 51.99% of all video games sold in 2013 making them the most purchased genre of game, and quite likely the most played genre (ESA, 2014). For both teacher education programs and current in-service teachers, that are incorporating game creation activities into their curriculum, it would be important to recognize that there may indeed be a predisposition to creating action games and in the case of males, violent games.

Examining video game playtime history, we found that overall males played more hours of games for every time period investigated than female pre-service teachers. These results are similar to Hayes and Ohrnberger (2013) who reported that male pre-service teachers played more hours of game while in university and in the past. We found a quadratic relationship for both genders in how much time pre-service teachers spent playing videos games. Both male and females spent more time playing video games in junior high and high school than they did in elementary and in recent years.

To gain a more complete understanding of the relationship between the number of hours pre-service teachers played per week on average and the period of their life (e.g., university, high school), a multinomial linear regression was used. The intervals (e.g., none at all, 1-3 hours) were compared for every time period (e.g., university, high school) to see if males or females were more likely to play more hours per week, using none at all as a baseline. While examining different periods of time, we found that in elementary school, males tend to play 4-6 or even more than 10 hours a week more than females, but our data did not find significant evidence for the 1-3 hour range or 7-10 hour range. For the 7-10 hour range, this could be because our dataset is not large enough or because at the elementary age, females could be more enthusiastic about games and spend more time playing them.

After elementary school, the amount of play time that females spend playing video games in a week seems to decrease. Males are more likely to play upwards of 4 hours of video gameplay in a week in junior high and high school in comparison to females, who will only play a couple hours a week for the same time period. In university, males are more likely than females to spend time playing video games. At this point of time, our data shows a decrease from 1-3 hours of game play to a preference of none at all for female video game playtime. In terms of video game usage, females will play video games when they are younger, but as they grow older, the amount of time they play games decreases.

Overall, we found that male pre-service teacher play considerably more hours of video games than females. This is consistent with the overall game play patterns found between males and females (Connolly, Boyle, Stansfield, & Hainey, 2007; Hainey, Boyle, Connolly, & Stansfield, 2011; Hayes and Ohrnberger, 2013) and has also been observed for the specific academic levels of elementary school (Annetta et al., 2009; Rideout et al., 2010; Greenberg et al., 2010), junior high school (Drummond & Dubow, 2011; Greenberg et al., 2010; Rideout et al., 2010), high school (Chou & Tsai, 2007; Drummond & Dubow, 2011; Greenberg et al., 2010; Rideout et al., 2010), and university (Rideout et al., 2010).

The social media history presented no significant gender differences with respect to time, but both genders were found to increase their usage of social media over time. A possible confounding variable could be that social media was gaining popularity as pre-service teachers that participated in our study grew older. Soomro, Kale, and Zai (2014) found that pre-service teachers use social media much more than in-service teachers. As a general benchmark comparison, 80% of K-12 teachers report using social media for personal or professional reasons (Bolkan, 2014) and more than 90% of college students have a profile on Facebook (Kaplan & Haenlein, 2010).

We also examined whether previous hours of experience with social media or playing video games had any correlation with genre of game created, as well as genders' past social media and video game playing habits and if these had an effect on the genre of game created. No relationship was found in either case, nor did we find anything correlating social media or video game play amount, and genre of game created in the literature.

Our study has a few limitations that should be addressed in future research. First, our sample is not representative of all pre-service teachers. While our sample consisted of both pre-service teachers in an elementary and secondary traditional four-year undergraduate teacher programs, this sample was based on those students who selected this educational technology course as an education course elective. Therefore they choose this class based on their interest in the area. Interestingly, with respect to the number of pre-service teacher males and females in our study, we had more males than females. The overall demographic of students in the pre-service teacher program at our post-secondary institution in our study are conversely 77 % female and 23 % male. Perhaps males are more interested in creating interactive multimedia. While our study has a considerably smaller amount of female participants, our results in relation to past video game experience repeat Hayes and Ohrnberger's (2013) findings which had fewer male participants. Additionally, Hayes and Ohrnberger (2013) also explored first year pre-service teachers, whereas we looked into pre-service teachers who were near the end of their program. In both studies, pre-service male teachers were found to play more hours of video games than pre-service female teachers.

On the subject of digital games, Gershenfeld (2011) succinctly wrote:

> Designing a digital game requires one to think analytically and holistically about games as systems, to experiment and test out theories, to solve problems, to think critically, and to effectively create and collaborate with peers and mentors. These are all skills that will be needed in a twenty-first century where virtually every job will involve navigating a complex, ever changing, digitally networked global landscape and where many of the future jobs have yet to be invented (p. 55).

References

Adams, E. (2007). *Fundamentals of game design*. Upper Saddle River N.J.: Pearson Prentice Hall.

AllGame. (2010). *Game Genres.* Retrieved from http://www.allgame.com/genres.php

Anderson, A. F., & Bavelier, D. (2011). Action game play as a tool to enhance perception, attention and cognition. *Computer games and instruction*, 307-329.

Annetta, L., Mangrum, J., Holmes, S., Collazo, K., & Cheng, M. T. (2009). Bridging Realty to Virtual Reality: Investigating gender effect and student engagement on learning through video game play in an elementary school classroom. *International Journal of Science Education*, *31*(8), 1091-1113. doi: 10.1080/09500690801968656

Apperley, T. (2006). Genre and game studies: Toward a critical approach to video game genres. *Simulation Gaming* 2006 37: 6. doi: 10.1177/1046878105282278

Baird, J. R., Fensham, P. J., Gunstone, R. F., & White, R. T. (1991). The importance of reflection in improving science teaching and learning. *Journal of research in Science Teaching*, 28(2), 163-182. doi: 10.1002/tea.3660280207

BELIEVE. (2014). Scriptease (Version 2.8) [computer software] Available from http://webdocs.cs.ualberta.ca/~script/

Boechler, P., Carbonaro, M., deJong, E., Stroulia, E. & Gutiérrez, L. (2011). Video games for teaching and learning: game-building as a teaching tool. In M. Koehler & P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2011* (pp. 2054-2061). Chesapeake, VA: AACE. Retrieved June 25, 2014 from http://www.editlib.org/p/36606

Boechler, P., Artym, C., DeJong, E., Carbonaro, M., & Stroulia, E. (2014). Computational thinking, code complexity and prior experience in a videogame-building assignment. *Proceedings of the IEEE International Conference on Advanced Learning Technologies,* Athens, Greece. doi: 10.1109/ICALT.2014.118

Boechler, P. M., Leenaars, L., & Levner, I. (2008). Recreational vs. educational computer experience: Predicting explicit and implicit learning outcomes in a website search. Las Vegas: Proceedings of the Society for Information Technology & Teacher Education International Conference.

Bolkan, J. (2014, Jan 15). Report: most teachers use social media, worry it will cause conflict. *The Journal*. Retrieved from http://thejournal.com/articles/2014/01/15/report-most-teachers-use-social-media-worry-it-will-cause-conflict.aspx

Bos, B., Wilder, L., Cook, M., & O'Donnell, R. (2014). Learning mathematics through Minecraft. *Teaching Children Mathematics*, *21*(1), 56-58. doi: 10.5951/teacchilmath.21.1.0056

Bourgonjon, J., Valcke, M., Soetaert, R., & Schellens, T. (2010). Students' perceptions about the use of video games in the classroom. *Computers & Education*, 54(4), 1145-1156. doi: 10.1016/j.compedu.2009.10.022

Carbonaro, M., Cutumisu, M., Duff, H., Gillis, S., Onuczko, C., Siegel, J., ... & Waugh, K. (2008). Interactive story authoring: A viable form of creative expression for the classroom. *Computers & Education*, 51(2), 687-707. doi: 10.1016/j.compedu.2007.07.007

Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: A gender-neutral attractor to computing science. *Computers & Education*, 55(3), 1098-1111. doi: 10.1016/j.compedu.2010.05.007

Chou, C., & Tsai, M. J. (2007). Gender differences in Taiwan high school students' computer game playing. *Computers in Human Behavior*, *23*(1), 812-824. doi: 10.1016/j.chb.2004.11.011

Clark, R. E., Yates, K., Early, S., & Moulton, K. (2011). An analysis of the failure of electronic media and discovery-based learning. In K. H. Silber & W. R. Foshay (Eds.), *Handbook of improving performance in the workplace* (pp. 263–297). San Francisco, CA: Pfeiffer. doi: 10.1002/9780470587089.ch8

Cohoon, J. M. (2006). Just get over it or get on with it: retaining women in undergraduate computing. In J. M. Cohoon, & W. Aspray (Eds.), *Women and information technology: Research on underrepresentation* (pp. 205–237). Cambridge MA: MIT Press. doi: 10.7551/mitpress/9780262033459.003.0007

Connolly, T.M., Boyle, E. A., Stansfield, M.H., & Hainey T. (2007). The potential of online games as a collaborative learning environment. *International Journal for Advanced Technology for Learning*, 4(4), 208-0923. Retrieved from http://www.actapress.com/Content_Of_Journal.aspx?JournalID=63

Denner, J., Werner, L., & Ortiz, E. (2011). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. *Computers & Education*, 58(1), 240-249. doi: 10.1016/j.compedu.2011.08.006

Drummond, J. A., & Dubow, E. F. (2011). Violent video games, aggressive behavior, and social/ school adjustment. Retrieved from http://www.education.com/reference/article/violent-video-games-social-school-aggressive/

Elliott, L., Ream, G., McGinsky, E., & Dunlap, E. (2012). The contribution of game genre and other use patterns to problem video game play among adult video gamers. *International journal of mental health and addiction*, 10(6), 948-969. doi: 10.1007/s11469-012-9391-4

ESA, the. (2013). *Demographic and usage data essential facts about the computer and video game industry*. Retrieved from http://theesa.ca/wp-content/uploads/2013/10/Essential-Facts-English.pdf

ESA, the. (2014). 2014 *Essential facts about the computer and video game industry*. Retrieved from http://www.theesa.com/wp-content/uploads/2014/11/ESA-Media-Resources.zip

Fischer, K. W., & Immordino-Yang, M. H. (2002). Cognitive development and education: From dynamic general structure to specific learning and teaching. Traditions of scholarship in education. *Chicago: Spencer Foundation*.

Ford, J. L. (2008). *Scratch Programming for Teens*. Course Technology PTR, Boston MA.

Franklin, T. & Annetta, L. (2011). Preface to Special Issue: Digital games and simulations in teacher preparation. *Journal of Technology and Teacher Education*, 19(3), 239-242. Retrieved from http://site.aace.org/pubs/jtate/default.html

GameFAQs. (2014). *Category Search*. Retrieved from http://www.gamefaqs.com/search/index.html?game=genre

GameFly, Inc. (2003, February 28). *Genre Definitions*. Retrieved from http://www.mobygames.com/glossary/genres/

Games for Entertainment and Learning Lab. (2014, July 29). *Games for entertainment and learning lab*. Retrieved December 4, 2014, from http://www.gel.msu.edu/

Gee, J. P. (2013). *Good video games and good learning: Collected essays on video games, learning and literacy* (2nd ed.). New York, NY: Peter Lang Publishing Inc.

Gershenfeld, A. (2011). Leveling up from player to designer: engaging and empowering youth through making video games. *Knowledge Quest*, *40*(1), 54-59. Retrieved from http://www.editlib.org/p/111220/

Good, J. (2011). Learners at the wheel: Novice programming environments come of age. *International Journal of People-Oriented Programming (IJPOP)*, *1*(1), 1-24. doi: 10.4018/ijpop.2011010101

Greenberg, B. S., Sherry, J., Lachlan, K., Lucas, K., & Holmstrom, A. (2010). Orientations to video games among gender and age groups. *Simulation & Gaming*, 41(2), 238-259. doi: 10.1177/1046878108319930

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. doi: 10.3102/0013189x12463051

Hainey, T., Boyle, E., Connolly, T., & Stansfield, M. (2011, June). Gender Differences in Motivations for Playing Computer Games: A Combined Analysis of Three Studies. In *Proceedings of the 7th European Conference on Management Leadership and Governance: ECGBL 2011* (p. 211). Academic Conferences Limited.

Hamlen, K. R. (2010). Re-examining gender differences in video game play: Time spent and feelings of success. *Journal of Educational Computing Research*, 43(3), 293-308. doi: 10.2190/ec.43.3.b

Hayes, E. (2008). Game content creation and it proficiency: An exploratory study. *Computers & Education*, 51(1), 97-108. doi: 10.1016/j.compedu.2007.04.002

Hayes, E. (2011). Gender and gaming. In J. D. Fletcher & S. Tobias (Eds.), *Computer games and instruction* (p. 453-476). IAP.

Hayes, E., & Ohrnberger, M. (2013). The gamer generation teaches school: The gaming practices and attitudes towards technology of pre-service teachers. *Journal of Technology and Teacher Education*, *21*(2), 154-177. Retrieved from http://www.editlib.org/p/41242/

Heeter, C., Egidio, R., Mishra, P., Winn, B., & Winn, J. (2008). Alien Games Do Girls Prefer Games Designed by Girls?. *Games and Culture*, 4(1), 74-100. doi: 10.1177/1555412008325481

Henry, C. (2011). *Genre Evolution in Video Games and a Framework for Analysis* (Doctoral dissertation, University of Alberta). Retrieved from http://hdl.handle.net/10048/2178

IBM. (2014). SPSS Statistics (version 21.0.0.0) [computer software]. Available from http://www-01.ibm.com/software/analytics/spss/

IGN. (2014). *Game Reviews*. Retrieved from ca.ign.com/games/reviews?filter=latest

Irish Computer Society. (2014). *Scratch teacher training*. Retrieved from http://www.scratch.ie/content/scratch-teacher-training

Jenkins, H., & Cassell, J. (2009). From Quake Grrls to Desperate Housewives: A decade of gender and computer games. *Beyond Barbie and Mortal Kombat: New perspectives on gender and gaming*, 5-20. doi: 10.5860/choice.46-6619

Jones, B., Valdez, G., Norakowski, J., & Rasmussen, C. (1994). Designing learning and technology for educational reform. *North Central Regional Educational Laboratory*.

Kafai, Y. B. (1996). Learning design by making games: Children's development of design strategies in the creation of a complex computational artifact. In Y. Kafai &M. Resnick (Eds.), *Constructionism in practice: Designing, thinking and learning in a digital world* (pp. 71–96). Mahwah, NJ: Lawrence Erlbaum Associates. doi: 10.4324/9780203053492

Kafai, Y. B., Franke, M. L., Ching, C. C., & Shih, J. C. (1998). Game design as an interactive learning environment for fostering students' and teachers' mathematical inquiry. *International Journal of Computers for Mathematical Learning*, 3(2), 149-184. doi: 10.1023/a:1009777905226

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. *Business horizons*, 53(1), 59-68. doi: 10.1016/j.bushor.2009.09.003

Kebritchi, M., & Hirumi, A. (2008). Examining the pedagogical foundations of modern educational computer games. *Computers & Education*, 51(4), 1729-1743. doi: 10.1016/j.compedu.2008.05.004

Kelleher, C. (2007). Storytelling Alice (version 2) [computer software]. Available from http://www.alice.org/kelleher/storytelling/

Kenny, R. F., & McDaniel, R. (2011). The role teachers' expectations and value assessments of video games play in their adopting and integrating them into their classrooms. *British Journal of Educational Technology*, 42(2), 197-213. doi: 10.1111/j.1467-8535.2009.01007.x

Klopfer, E., Osterweil, S., Groff, J., & Haas, J. (2009). Using the technology of today, in the classroom today. The instructional power of digital games, social networking, simulations, and how teachers can leverage them. *The Education Arcade*, 1-21. Retrieved from http://education.mit.edu/

Lee, M., Ko, Y., Song, H., Kwon, K., Lee, H., Nam, M., & Jung, I. K. (2007). Characteristics of internet use in relation to game genre in Korean adolescents. *CyberPsychology & Behavior*, *10*(2), 278-285. doi: 10.1089/cpb.2006.9958

Lenhart, A., Jones, S., & Macgill, A.R. (2008a). *Adults and video games*. Washington, DC: Pew Research Center. Retrieved from http://www.pewinternet.org/2008/12/07/adults-and-video-games/

Lenhart, A., Kahne, J., Middaugh, E., Macgill, A. R., Evans, C., & Vitak, J. (2008b). *Teens, video games, and civics: teens' gaming experiences are diverse and include significant social interaction and civic engagement*. Pew Internet & American Life Project.

Lucas, K. & Sherry, J. (2004). Sex differences in video game play: A communication-based explanation. *Communication Research*. (October 2004), 31 (5), p. 499-523. doi: 10.1177/0093650204267930

Mayer, R.E. (2014). *Computer games for learning: An evidence-based approach*. Cambridge, MA: The MIT Press.

Microsoft. (2014). Kodu Game Lab (Version 1.4.59.0) [computer software]. Available from http://www.kodugamelab.com/

Millstone, J. (2012, May). Teacher attitudes about digital games in the classroom. In *The Joan Ganz Cooney Center at Sesame Workshop*.

Mishra, P., & Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *The Teachers College Record*, 108(6), 1017-1054. doi: 10.1111/j.1467-9620.2006.00684.x

MIT. (2009). Scratch (Version 1.4) [computer software]. Available from http://scratch.mit.edu/

Mumtaz, S. (2000). Factors affecting teachers' use of information and communications technology: a review of the literature. *Journal of information technology for teacher education*, 9(3), 319-342. doi: 10.1080/14759390000200096

Papert, S. (1971). A Computer Laboratory for Elementary Schools. *Computers and Automation*. Retrieved from http://dspace.mit.edu/handle/1721.1/5834

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S. (1991). Situating Constructionism. In I. Harel & S.Papert (Eds.), *Constructionism* (pp. 5-23). Norwood, NJ:Ablex.

Papert, S. (1998). Does easy do it? Children, games, and learning. *Game Developer*, 5(6), 88. Retrieved from http://www.gdcvault.com/gdmag

Penner, D. E. (2000). Cognition, computers, and synthetic science: Building knowledge and meaning through modeling. *Review of research in education*, 1-35. doi: 10.3102/0091732x025001001

Phan, M. H., Jardina, J. R., Hoyle, S., & Chaparro, B. S. (2012, September). Examining the role of gender in video game usage, preference, and behavior. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 56, No. 1, pp. 1496-1500). SAGE Publications. doi: 10.1177/1071181312561297

Piaget, J. (1970). *Genetic epistemology*. Trans. E. Duckworth. doi: 10.1177/000276427001300320

Pierson, M. E. (2001). Technology integration practice as a function of pedagogical expertise. *Journal of Research on Computing in Education*, *33*(4), 413-430. doi: 10.1080/08886504.2001.10782325

Prensky, M. (2008). Students as designers and creators of educational computer games: Who else?. *British Journal of Educational Technology*, 39(6), 1004-1019. doi: 10.1111/j.1467-8535.2008.00823_2.x

Rideout, V. J., Foehr, U. G., & Roberts, D. F. (2010). *Generation M2: Media in the lives of 8–18 year-olds*. Kaiser Family Foundation. January 2010. <http://www.kff.org/entmedia/upload/8010.pdf> Retrieved 10.05.11.

Robertson, J. (2012). Making games in the classroom: Benefits and gender concerns. *Computers & Education*, 59(2), 385-398. doi: 10.1016/j.compedu.2011.12.020

Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computers & Education*, 50(2), 559-578. doi: 10.1016/j.compedu.2007.09.020

Rollings, A., & Adams, E. (2003). *Andrew Rollings and Ernest Adams on game design*. New Riders.

Salen, K. (2007). Gaming literacies: A game design study in action. *Journal of Educational Multimedia and Hypermedia*, 16(3), 301-322. Retrieved from http://www.editlib.org/p/24374

Salen, K. (2011). *Quest to learn: Developing the school for digital kids*. MIT Press.

Salerno, K. (2013). Video Games and Learning: Teaching and Participatory Culture in the Digital Age, by Kurt Squire. *Alberta Journal of Educational Research*, 59(1), 129-132. Retrieved from http://www.ajer.ca/

Schmid, R.F., Bernard, R.M., Borokhovski, E., Tamim, R.M., Abrami, P.C., Surkes, M.A., Wade, C.A., & Woods, J. (2014). The effects of technology use in postsecondary education: A meta-analysis of classroom applications. *Computers & Education*, *72*, 271-291. doi: 10.1016/j.compedu.2013.11.002

Schrader, P. G., Archambault, L. M., & Oh-Young, C. (2011). Training by Gaming: Preparing Teachers of Today for Tomorrow's Learning Environments. *Journal of Technology and Teacher Education*, 19(3), 261-286. Retrieved from http://www.editlib.org/p/35993/

Schrader, P. G., Zheng, D., & Young, M. (2006). Teachers' perceptions of video games: MMOGs and the future of preservice teacher education. *Innovate: Journal of Online Education*, 2(3), 1-10. Retrieved from http://www.nyu.edu/classes/keefer/waoe/waoej.html

Shapiro, J., SalenTekinbas, K., Schwarts, K., & Darvasi, P. (2014). *The MindShift Guide to Digital Games and Learning* [PDF]. Available from http://www.kqed.org/assets/pdf/news/MindShift-GuidetoDigitalGamesandLearning.pdf

Smith, C.P., & Neumann, M.D. (2014). Scratch it out! Enhancing geometrical understanding. *Teaching Children Mathematics*, *21*(3), 185-187. doi: 10.5951/teacchilmath.21.3.0185

Soomro, K. A., Kale, U., & Yousuf Zai, S. (2014). Pre-service teachers' and teacher-educators' experiences and attitudes toward using social networking sites for collaborative learning. *Educational Media International*, (ahead-of-print), 1-17. doi: 10.1080/09523987.2014.977003

Steinkuehler, C., & Squire, K. (2014). *Video Games and Learning*. Retrieved from: https://www.coursera.org/course/videogameslearning

Tamin, R.M., Bernard, R.M., Borokhovski, E., Abrami, P.C., & Schmid, R.F. (2011). What forty years of saya about the impact of the technology on learning: A second-order meta-analysis and validation study, *Review of Educational Research*, *81*(1), 4-28. doi: 10.3102/0034654310393361

Tobias, S., & Fletcher, J. D. (Eds.). (2011). *Computer games and instruction*. IAP. doi: 10.1111/bjet.12045_2

Ventura, M., Shute, V. & Jeon, K. (2012). Video game play, personality and academic performance. *Computers & Education*. May 2012; 58 (4), p. 1260-1266. doi: 10.1016/j.compedu.2011.11.022

Vos, N., van der Meijden, H., & Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education*, 56(1), 127-137. doi: 10.1016/j.compedu.2010.08.013

Whitton, N. (2014). *Digital Games and Learning: Research and Theory*. London, UK: Routledge. doi: 10.4324/9780203095935

Williams, M.K., Foulger, T.S. & Wetzel, K. (2009). Preparing pre-service teachers for 21st century classrooms: Transforming attitudes and behaviors about innovative technology. *Journal of Technology and Teacher Education*, 17(3), 393-418. Retrieved from http://www.editlib.org/p/28216/

Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi: 10.1145/1118178.1118215

Wing, J. (2009). Proceedings from Computer Science Education Conference. Paris, France. doi: 10.1109/iccse.2009.5228576

Wolf, M. J. (2008). *The video game explosion*. Greenwood press, 88 post road west, Westport: Greenwood Publishing Group,Inc.

Young, M. F., Slota, S., Cutter, A. B., Jalette, G., Mullin, G., Lai, B., ... & Yukhymenko, M. (2012). Our Princess Is in Another Castle A Review of Trends in Serious Gaming for Education. *Review of educational research*, *82*(1), 61-89. doi: 10.3102/0034654312436980

Appendix:

Computer Experience Questionnaire

VIDEO GAMES
Q: Please indicate how much time you spent PLAYING VIDEO GAMES, on average, PER WEEK, during the following:

- In recent weeks
- While in High School
- While in Junior High
- While in Elementary

On the Likert scale

- Not at all
- 1-3 Hrs
- 4-6 Hrs
- 7-10 Hrs
- >10 Hrs

SOCIAL NETWORKING
Q: Please indicate how much time you spent USING SOCIAL NETWORKING (e.g. Facebook, Twitter, MSN, E-mail), on average, PER WEEK, during the following:

- In recent weeks
- While in High School
- While in Junior High
- While in Elementary

On the Likert scale

- Not at all
- 1-3 Hrs
- 4-6 Hrs
- 7-10 Hrs
- >10 Hrs

Evaluating Computational Thinking in Digital Games Constructed by Pre-service Teachers

Abstract

This paper details the creation and trial of the Quality Practices of Game Design Survey (QPDGS), an instrument created to measure the skills pre-service teachers demonstrated in their development of a digital game in *Scratch* (MIT, 2009). The QPDGS was derived through a literature review in the area of game design and computational thinking. The survey consists of 28 items grouped into key areas: Problem Solving, Computational Thinking, Customization of Player Experience, Player Interaction, Player Immersion, Player Motivation, and Interface Usability. In this paper, the QPDGS was used to evaluate a purposeful, representative sample (*n*=40) of the 166 pre-service teachers who created a digital game. Frequencies were found in evaluations and items were compared in the form of a correlational matrix. Overall, the set of video games built by the pre-service teachers indicate partial, but not complete, awareness of computational and game design principles. The QPDGS may be valuable for developing an assessment tool for computational thinking and game design skills outside of *Scratch*.

Keywords: digital games, video games, computational thinking, game design, pre-service teacher education, Scratch, learning, technology

In 1962, Alan Perlis, the first recipient of the ACM A.M. Turing Award, argued the necessity for college students to learn programming and the “theory of computation” (p. 195) to enhance their understanding of their separate disciplines (Guzdial, 2008). This “theory of computation” has been taught under the label of Computing/Computer Science (CS) for 50 years and in 2003, the ACM Model Curriculum for K-12 Computer Science distinctly defined CS as “the study of computers and algorithmic process including their principles, their hardware and software design, their applications, and their impact on society” (Tucker et al., p. 1). In 2006, Jeannette Wing, the corporate vice president at Microsoft Research and former President's Professor of Computer Science at Carnegie Mellon University, repurposed the CS concepts Perlis emphasized as Computational Thinking (CT) specifying “computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (p. 33). Wing asserts that CT is a skill everyone should have, further creating the distinction: CT is not synonymous with programming or CS. Established aspects of CT include:

> 1) Solving problems and designing systems that draws on concepts fundamental to computer science;
> 2) Creating and employing different levels of abstraction, to understand and solve problems more effectively;
> 3) Thinking algorithmically and with the ability to apply mathematical concepts to develop more efficient, fair, and secure solutions; and
> 4) Understanding the consequences of scale, not only for reasons of efficiency but also for economic and social reasons (Lu, & Fletcher, 2009, p. 1).

In a 2006 workshop Blum and Cortina (2007) found that teachers believed CT was indistinguishable from programming; by the end of the workshop the teachers’ understanding had shifted from focusing on “problem solving/algorithms” and “learning using programming languages” to “developing computational thinking skills for all aspects of life” and “problem solving techniques/algorithms” (p. 22). Pre-service teachers will one day educate the professionals of tomorrow and it is important these teachers are equipped with the skills to do so. CT can be taught through the construction of digital game artifacts. Incidentally, effective digital games employ the best educational practices making Game Design (GD) an even more beneficial skill for pre-service teachers to acquire (Gee, 2013). In this study pre-service teachers were provided with the opportunity to construct a digital game of their choice as part of a pre-service educational technology course. After these games were produced, they were measured on the *Quality Practices of Game Design Survey* (QPGDS), which measures their proficiency in GD and CT. These skills can potentially transfer into pedagogical practice in the classroom to prepare 21st century learners.

The next sections of the paper outline: A) why CT and GD should be taught in a K-12 classroom; B) how they can be taught through the construction of digital games; C) details the creation of the QPGDS; D) Explains the structure of the study; E) Explores the QPGDS with a purposeful dataset; F) Summarizes the results; and G) Discusses the findings.

CT in Education

Pre-service teachers constructing the physical artifact of a digital game utilize the educational theory of constructionism. In 1954, Piaget formalized constructivism, whereby children actively construct knowledge through experience. In 1967, Papert designed Logo to demonstrate the practice of constructionism with the intention of developing procedural thinking through programming (Papert, 1971; Paper 1980).

Constructionists approach knowledge development by emphasizing the concrete rather than the abstract and believe that manipulating tangible objects aids the process of knowledge demonstration (Carbonaro, 1997). The power of constructionism comes from the creation of a physical, personally meaningful product representing a person’s knowledge, which affords a custom experience where students actively take ownership for their learning (Bruckman & Resnick, 1995; Chambers & Carbonaro, 2003). Constructionism asserts learners as active participants in the learning process rather than passive passengers receiving knowledge from an instructor (Kafai & Resnick, 1996).

Programming allows learners to take on an active role as the act of programming is the construction of code and is thereby also constructionism in practice. Lu and Fletcher (2009) write that "programming is to Computer Science what proof construction is to mathematics and what literary analysis is to English" (p. 261) acknowledging that traditional CS is not taught without programming, but there is a need for CT skills outside of CS. In response, Lu and Fletcher (2009) have produced the Computational Thinking Language including major elements of programming: state, data, iteration, searching, and efficiency. Hemmendinger (2010) clarified that the goal of teaching CT is not for everyone to think like a computing scientist, but rather for them to think like an economist, physicist, or artist by "understand[ing] how to use computation to solve their problems, to create, and to discover new questions that can fruitfully be explored" (p. 6).

Why should CT be taught?

The President's Information Technology Advisory Committee acknowledge the impact of CT and the skill sets' widespread benefits writing "Computational science – the use of advanced computing capabilities to understand and solve complex problems – has become critical to scientific leadership, economic competitiveness, and national security" (2005, p. iii); the report continues by saying "[t]he PITAC believes that computational science is one of the most important technical fields of the 21st century because it is essential to advances throughout society" (p. iii).

In order to advance society, CT should be introduced into K-12, but in order to teach it to our K-12 students, teachers need to be well-versed in CT themselves. Unfortunately, students and pre-service teachers rarely have an opportunity to be immersed in CT (through CS) until they reach post-secondary studies, but Tai, Liu, Maltese, and Fan (2006) found that students often decide whether they want to pursue a career in the sciences by the eighth grade. Without early exposure to CT, how do students know that it is even an option? In order to cultivate a generation with CT skills, students need to be familiarized with CT throughout their K-12 education (Barr and Stephenson, 2011).

CT skills have traditionally been acquired through programming. Traditionally when learning a new programming language, one of the first tasks attempted is to print the message ‘Hello World!’ Below the task is completed in C:

```
#include <stdio.h>
main() {
printf("Hello, World!\n");
}
```

One criticism of beginning to learn CT through programming is that students may need an extensive knowledge about “libraries, main functions, functions, strings, control characters, and programming language syntax before they can achieve this rather simple and not very exciting output” (Federici & Stern; 2011, p. 1353). As the means are complex and the result is not overly meaningful, it can be an obstacle for teaching CT skills. Bers, Flannery, Kazakoff, and Sullivan (2014) demonstrated that when material is appropriately scaffolded, even students as young as kindergarteners can be motivated to learn aspects of robotics, programming, and CT. Papert (1980) qualified these scaffolding tools as low floor (easy to learn) and high ceiling (capable of sophisticated programs) and kept these principles in mind with the creation of LOGO. Even though CT is contemporarily, and almost exclusively, introduced at the post-secondary level, using low floor, high ceiling tools, core CT concepts and capabilities can be applied throughout curriculum in kindergarten through grade twelve classrooms (Barr & Stephenson, 2011). The more time middle school-aged students spent working in a programming environment, the more complex their projects became (Anton, Harris, Ochsner, Rothschild, & Squire, 2013). There is a wealth of visual coding environments available for younger students such as *Scratch* (MIT, 2009) or *Kodu* (Microsoft, 2014). If children start working on these skills early, they can be well-developed for their careers.

Jeannette Wing, speaking at the Columbia Journalism School (2014), discussed how CT is transferable to all professions. She focused on the application of abstraction and how the understanding of the concept is universally beneficial. Forgetting about the details of any circumstance allows for a change of focus allowing humans to focus their somewhat limited computing power (when compared to that of a computer). This is highly practical in problem-solving.

CT, comparable to critical or mathematical thinking, is one approach or framework that can be used for solving problems that involve high-level elements of: confidence and persistence in solving complex problems, acceptance of ambiguity, ability to work with open-ended problems, and the ability to communicate and work with others to attain a common goal (Barr, Harrison, & Conery, 2011). Elements of CT have a direct correspondence with elements of scientific inquiry (e.g., iterative and incremental development in the formal software engineering process can be applied to iterative theory model refinement or encapsulation of functions and states within an object as a means of abstraction can be applied to creating a complete, formal representation of scientific processes (Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013)). Wing (2006) has even proposed adding CT to every child’s analytical ability adding onto the three 'r' of reading, ‘riting, and ‘rithmetic. Grover and Pea (2013) also found this suggestion in media and industry and proposed ‘rithms, short for algorithm, be added as the fourth ‘r.’

Case Studies of CT in K-12 Education

Another important component of CT is abstraction. Abstraction, was linked to Piaget’s fourth stage of development: the formal operational stage (entered around age 12 and continued until adulthood), where children developmentally work on the acquisition of formal logic (Kramer, 2007); even though this stage is entered at age 12, children could be exposed to CT topics at an earlier age, but traditionally CT is taught through CS, which is not offered for most K-12 students.

Alberta. Looking locally in Alberta, CS is not a core science in primary or secondary education. In Alberta Education’s (2014) K-12 Programs of Study, CS falls under Career and Technology Studies (CTS) as a “complementary program” that students can choose as an elective.

Examining the University of Alberta Faculty of Science’s admission requirements, students must have credit in two 30 level high school sciences. In spring 2009, the University of Alberta added CS to the list of entry sciences, followed shortly by the University of Calgary, and the University of Lethbridge, but it has not been established as a science in Albertan high schools (Department of Computing Science, 2014). In addition, the Iverson CS Exam was also created to test CS proficiency and provide entrance scholarships to the Faculty of Science at the University of Alberta.

The United States. Code.org (2014), a website for promoting the education of CS in the United States contains testimonials from many celebrities and influential figures including Tony Cardenas, U.S. Congressman of California, “The tech industry is quickly expanding and adding jobs to the U.S. economy. We need a workforce trained with the computer science and coding skills that fuel this expansion, so we can keep these exciting jobs here." California, home of Silicon Valley which houses many of the world's leading technology corporations, has more reasons than others to promote CS in K-12 education. The State Government of California (2014) has recognized and acted on the importance of CS to the prosperity of their state making them a leader in the area of CS in education; Table 2.1 shows the status of proposed bills.

Table 2.1
Proposed and passed CS bills in California.

Bill	State of Bill	Description
AB 1530	Draft	Bill would add CS to the required program of study for grades 1 through 6.
AB 1539	Passed	The California Board of Education will adopt CS standards for grades 7 through 12.
AB 1540	Draft	Bill would allow high school students to receive CS credit through local community colleges.
AB 1764	Passed	School boards are authorized to award students a third year math credit for satisfactorily completing a CS course.
AB 2110	Draft	Bill would require the California Board of Education to include CS content in existing curriculum frameworks.
SB 1200	Passed	Governing boards of the public higher education systems will establish academic standards for high school CS courses that would be acceptable at post-secondary institutions.

The United Kingdom. The Royal Society in the United Kingdom found the delivery of computing education "highly unsatisfactory" (Furber, 2012, p. 5) citing a shortage of technologically and computationally competent teachers resulting in classrooms where non-specialists end up teaching the content. In addition, the problem is not improving: there is a lack of professional development in the area of Computing. The Royal Society acknowledges the need to have infrastructure for these teachers to impart to their students and attributes the lack of interest in senior level high school CS to the lack of demand from higher educational institutions.

The report identifies the following countries that are actively reforming their high school CS curricula: United States, Israel, New Zealand, Germany, India, and South Korea. Most importantly, the report identified that computational thinking is pervasive in other disciplines and should be an encouraged competency for all students.

CT Taught Through Gaming

One topic that appears frequently in the Royal Society of the United Kingdom’s report on CT educational reform is ‘digital gaming.’ Even without the digital component, gaming is an exceptional vehicle for teaching CT skills. Gee (2007) draws attention to *Yu-Gi-Oh* and *Magic: the Gathering* trading card games. These trading cards can be used to engage in battles with other collectors where there is an infinite amount of permutations of conditions and combinations of cards making conditional logic and algorithmic thinking crucial skills. In one turn in a *Yu-Gi-Oh* card battle, one player could have its stats raised by two condition-based cards, they could attack their opponent, have their attack redirected by an opponent’s condition-based trap card, have that attack re-re-directed by one of their own conditional trap cards, and finally negated by another condition-based trap card depending on the finite actions they invoke on their turn. The game can quickly get complicated, but players master the conditional statements and quickly understand new conditions played by their opponents.

Board game creation and play could also be a medium for CT skill development in younger children. The non-digital component is more affordable for schools that do not have access to computers for all students. The game mechanic for *Rock, Paper, Scissors* depends on the conditions if player A plays rock, and player B plays scissors, then player A wins. Similar games can easily be created by students. The board game *Pandemic* has been cited as a game that allows players to develop conditional logic, algorithm building, debugging, simulation, and distributed computation skills (Berland & Lee, 2011). Through play and game design (GD), children can reinforce proficiency with CT.

Adding the digital medium can increase the level of complexity of gaming. Gee (2013) advocates the usage of video game creation for systems thinking and CT skill development. Instead of checking a single condition by the human brain at a time, a computer can complete multiple operations. Basawapatna, Koh, and Repenning (2010) had middle students create games, develop CT skills, and engage others who played their games by transferring complex, diverse concepts. In a later study, Basawapatna, et al. (2011), saw a transfer of their students' CT skills to other contexts from GD.

Boechler, Artym, DeJong, Carbonaro, and Stroulia (2014) analyzed games created by pre-service teachers and found that those who played more hours of digital games actually used a higher number of variables in their games, resulting in more complex games. They go on to suggest that the more experience pre-service teachers have in playing digital games, the more these teachers may grasp the mechanics that underlay digital games, including the skills involved in CT, and the more this understanding appears in the game they design.

Gee (2013) and Rajaravivarma (2005) stress the low cost of failure associated with digital games. Usually digital games have a set number of lives, checkpoint, or game over component allowing them to try multiple times without permanent, adverse effects. This 'low cost of failure' allows a student to attempt the game without having to worry about lasting consequences. Factors like low cost of failure are also good practices of education.

Video game development is an engaging area that can get pre-service teachers, and their students, excited about learning important CT programming skills. Harel and Papert (1991) maintain that the challenges of programming can be easier to learn when the tasks are interesting.

There are many conventions of game development that pre-service teachers may not explicitly be aware of such as challenge levels, goals and sub-goals, interactions, and rules.

In the near future, CT will impact everyone, in every field, and will be a fundamental skill used by everyone (Wing, 2006). GD is a developing, but well-received, vehicle that can be used to teach CT; additionally, it can help educators learn good educational practices that can help them be effective teachers (Gee, 2013). Today's classrooms must include a fusion of technological fluency and content preparing students to work and live in the Information Age (Denton, 2012).

Assessment of Skills

Using programming can help promote the development of CT skills that are increasingly important for problem solving in today's world; furthermore GD is an excellent way to learn these CT skills (Good, 2011; Grover & Pea, 2013; Papert, 1980; Riley & Hunt, 2014; Wing, 2006). Gee (2013) illustrates that GD incorporates the best practices of education making it a valuable skill for educators to have. As educators are teaching the professionals of tomorrow, they must first have the skills they wish to impart. So far, this paper has proposed that CT and GD are important skills for teachers to possess.

First, what are the specific competencies teachers should have? Before they can be assessed, the specific skills must first be identified. Through an exhaustive literature review, the following areas critical to CT and GD were identified.

Computational Thinking

A comprehensive literature review on CT was completed citing the following authors: Perlis (1962), Knuth (1985), NRC (1999), Wing (2006), Kramer (2007), Ater-Kranov, Bryant, Orr, Wallace, & Zhang (2010), Google (2010), Barr & Stephenson (2011), (ISTE; 2011), Brennan & Resnick (2012), Grover & Pea (2013), Sengupta, Kinnebrew, Basu, Biswas, & Clark (2013), and Touretzky, Marghitu, Ludi, Bernstein, Ni (2013). Components and descriptions are noted if they were explicit in the source article. Denning (2009) made the argument that CT developed from algorithmic thinking in the 1950s and 1960s. In 1962, Perlis outlined fundamentals of programming, which also incidentally outline some of the fundamentals of CT. In 1985, Knuth identified the accepted components of algorithmic thinking at the time. The Committee on Information Technology Literacy (NRC; 1999) distinguish the accepted components of algorithmic thinking. From the review, the following table was derived (*See Table 2.2*).

Table 2.2
Computational thinking's main components by year, then by author.

Author and Year	Component	Description	Identifier
Perlis (1962)	Definitions	Reuse of useful code sequences.	CT01
	Iteration	Cycles that are ran with different data.	CT02
	Mechanical language	Being able to translate natural language and commands into a programming syntax.	CT03
	Parameterization	Required in each program.	CT04
	Priori attention to eventualities regardless of likelihood	Attention to the unlikeliest of cases.	CT05
	Proof	A programmer should be able to show that their program does what it is supposed to do.	CT06
	Recursion	Instructional sequences calling themselves.	CT07
	Representation	Choosing the optimal way to represent data.	CT08
	Simulation	Processes not relating to computers should be able to be modeled on a computer.	CT09
Knuth (1985)	Abstract reasoning		CT10
	Algorithms	Concepts dealing with well-defined processes.	CT11
	Formula manipulation	Involves pattern recognition.	CT12
	Information structures		CT13
	Reduction to simpler problems		CT14
	Representation of reality		CT15
NRC (1999)	Algorithm vs. program		CT16
	Basic data organizations	Record, array, list.	CT17
	Functional decomposition		CT18
	Generalization and parameterization		CT19
	Refinement		CT20
	Repetition	Iteration and/or recursion.	CT21
	Top-down design		CT22
Wing (2006)	Abstraction	Overlooking the details of an operation and focusing on higher-level meaning.	CT23
	Automation	Making a process optimal and self-sufficient.	CT24
Kramer (2007)	Abstraction	Changing focus, leaving out consideration of properties in a common object, withdrawing or removing something.	CT25
Ater-Kranov, Bryant, Orr, Wallace, & Zhang (2010)		Applying algorithmic approaches to analyze computation process.	CT26
		Applying reasoning to algorithmic and heuristic solutions to problems.	CT27

	[Recursion]	Applying recursive thinking.	CT28
	[Algorithmic thinking]	Breaking down a task into a series of distinct steps.	CT29
	[Generalization]	Generalizing a single observation or solution to explain multiple instances.	CT30
	[Optimization]	Identifying and weighing cost-benefit trade-offs of different solutions.	CT31
		Implementing and then interpreting a solution from the original problem.	CT32
	[Mathematical modeling]	Mathematic modeling.	CT33
		Modeling problems and calculating solutions.	CT34
		Understanding limitations of generalized solutions.	CT35
		Using critical thinking.	CT36
		Using logic, mathematics, and abstraction to solve problems.	CT37
		Using logical processes with computing technology.	CT38
Google (2010)	Algorithm design	Creating a step-by-step strategy for solving a problem.	CT39
	Decomposition	Being able to break a task or action into its most basic parts so it can be understood by another; can help with pattern recognition and generalization.	CT40
	Pattern recognition	Noticing differences and similarities in order to make predictions or find optimal scenarios; provides a basis for solving problems and designing algorithms.	CT41
	Pattern generalization and abstraction	Ignoring information that is not necessary to solve a problem.	CT42
Barr & Stephenson (2011)	Abstraction	Shift from concrete to general for solutions that are developed.	CT43
	Algorithms and procedures		CT44
	Automation		CT45
	Data analysis		CT46
	Data collection		CT47
	Data representation		CT48
	Parallelization		CT49
	Problem decomposition	Splitting problems into smaller components that may be easier to solve.	CT50
	Simulation		CT51
International Society for Technology in Education (ISTE; 2011; p. 1)	<u>Characteristics</u>		
	[Automation and algorithmic thinking]	"Automating solutions through algorithmic thinking (a series of ordered steps)."	CT52
	[Automation]	"Formulating problems in a way that enables us to use a computer and other tools to help solve them."	CT53
	[Optimization]	"Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources."	CT54
	[Data Analysis]	"Logically organizing and analyzing data."	CT55
	[Abstraction and Simulation]	"Representing data through abstractions such as models and simulations."	CT56
	[Generalization]	"Generalizing and transferring this problem solving process to a wide variety of problems."	CT57
	<u>Essential dimensions</u>		
		"Confidence in dealing with complexity."	CT58
		"Persistence in working with difficult problems."	CT59
		"The ability to communicate and work with others to achieve a common goal or solution."	CT60
		"The ability to deal with open ended problems."	CT61
		"Tolerance for ambiguity."	CT62
Brennan & Resnick (2012)	<u>Computational Concepts</u>		
	Conditionals	Carrying out a different outcome based on certain conditions.	CT63
	Data	The storing, retrieving, and updating of values.	CT64
	Events	One thing triggers another thing to happen.	CT65
	Loops	Used to run sequences multiple times.	CT66
	Operators	Functions that allow for the manipulation of data.	CT67
	Parallelism	When sequences of commands happen simultaneously.	CT68
	Sequences	A series of instructions that is executable by a computer.	CT69
	<u>Computational Perspectives</u>		
	Connecting	Having a support system for learning a new medium.	CT70
	Expressing	Being able to express ideas in a new medium.	CT71
	Questioning	Asking how new mediums can be used to express ideas.	CT72

	Computational Practices		
	Abstracting and modularizing	Using smaller collections of code to make a bigger project.	CT73
	Being incremental and iterative	Building onto a program little, by little.	CT74
	Reusing and remixing	Extending a project using other people's work.	CT75
	Testing and debugging	Troubleshooting strategies used through the programming process.	CT76
Grover & Pea (2013)	Abstractions and pattern generalizations (including models and simulations)	"Defining patterns, generalizing from specific instances" and required to deal with complexity (Wing, 2011).	CT77
	Algorithmic notions of flow of control		CT78
	Conditional logic		CT79
	Debugging and systematic error detection		CT80
	Efficiency and performance constraints		CT81
	Iterative, recursive, and parallel thinking		CT82
	Structured problem decomposition (modularizing)		CT83
	Symbol systems and representations		CT84
	Systematic processing of information		CT85
Sengupta, Kinnebrew, Basu, Biswas, & Clark (2013)	Algorithm design and complexity analysis		CT86
	Class inheritance and polymorphism	Being able to reuse properties within a class.	CT87
	Classes or agent-breed	Categories of object, with attributes, that can be instantiated, changes states, and act independently.	CT88
	Decentralized and multi-agent systems	Distributed intelligence.	CT89
	Encapsulation	Hide function and state within an object providing an abstraction while omitting details.	CT90
	Software engineering process	Incremental and iterative development.	CT91
	Unit testing		CT92
Touretzky, Marghitu, Ludi, Bernstein, Ni (2013)	Conceptual mapping	Described in terms of *Alice*, *Kodu* and *NXT-G*.	
	Conditionals	*Alice* uses the traditional if/then statements. *Kodu* uses When/do statements. *NXT-G* uses switch blocks with if/then/else.	CT93
	Looping	Alice uses *while* loop and a *loop n* times structures. *Kodu* uses implicit looping on all code lines unless the *once* tile is added. *NXT-G* uses a *loop* block.	CT94
	Objects	*Alice* and *Kodu* both have objects.	CT95
	Parallelism	*Alice* and *NXT-G* support parallel execution, but *Kodu* does not as actions take effect sequentially.	CT96
	States	In *Alice*, all statements can be regarded as a state. In *Kodu*, states are identified using pages of code on an object. *NXT-G*'s graphical layout is close to a state machine, but *wires* and *ports* are used to send parameters.	CT97
	Variables	*Alice* and *NXT-G* use variables. *Kodu* uses *scores* to represent variables.	CT98

Game Design

A systematic literature review was completed on GD citing the following authors: Dickey (2005), Gee (2005), Mayer (2011), and Gee (2013). Good practices of education create deep understanding and empower learners through problem-based learning, and are incidentally present in good video games (Gee, 2013). From the review, the following table was derived (*See Table 2.3*).

Table 2.3
Game Design Principles by year, by author.

Author and Year	Component	Description	Identifier
Dickey (2005)	Affiliation with others	I.e., role playing, non-player characters.	GD1
	Affirmation of performance	I.e., hooks.	GD2
	Challenging tasks		GD3
	Choice		GD4
	Clear & compelling standards		GD5
	Focused goals		GD6
	Novelty & variety	I.e., narrative arcs.	GD7
	Protection from adverse consequences for initial failures		GD8
Gee (2005)	Agency	Players can feel in control over what they are doing.	GD9
	Challenge and consolidation	Mastery learning with increasingly difficult problems to increase the level of a player's mastery.	GD10
	Cross-functional teams	Being able to work in teams with players of different specializations and understanding what other players can contribute.	GD11
	Customization	The player can customize their experience to fit their learning style, preference, or change difficulty.	GD12
	Explore, think laterally, rethink goals	Investigating the environment instead of taking the quickest way through a level.	GD13
	Identity	Players assume or create a strongly formed and appealing character.	GD14
	Interaction	The game should react and be an active experience.	GD15
	"Just-in-time" and "on demand"	In order to not overwhelm a player, information is provided as they need it or is accessible when they want it.	GD16
	Performance before competence	Having the opportunity to perform before being deemed competent on a subject.	GD17
	Pleasantly frustrating	Not too hard, but not too easy: providing the proper amount of challenge to motivate players.	GD18
	Production	Players co-design the game experience through their choices.	GD19
	Risk Taking	Consequences of failure are lowered.	GD20
	Situated meanings	Attaching meaning to words in different actions, images, or dialogues.	GD21
	Smart tools and distributed knowledge	Abstraction of game characters functions so that the player only needs to know certain things to effectively control them.	GD22
	System thinking	Players must consider relationships, events, facts, and skills collectively and the consequences of their actions.	GD23
	Well-ordered problems	Solutions to early problems give hints to solutions to later problems.	GD24
Mayer (2011)	Rule-based	Players can come to understand their environments.	GD25
	Responsive	In-game environments respond to the player's actions.	GD26
	Challenging	Game environments pose goals that are achievable and increase in difficulty.	GD27
	Cumulative	In-game environments keep records of player's past successes.	GD28
Gee (2013) [1]	Deep Understanding		
	Meaning as action	"Situated meaning": giving a context to words by images, actions, and experiences.	GD29
	Systems thinking	Comprehending how many variables interact with one another.	GD30
	Empowered Learners		
	Co-design principle	What a player creates matters.	GD31
	Customization	Players have the opportunity to try different styles, change the difficulty and the cost of failure is low.	GD32
	Identity	Games provide the opportunity for creating a strong character identify.	GD33
	Manipulation	Games afford the player control over the world they are in.	GD34
	Problem-based Learning		
	Cycles of expertise	A cycle of challenge, practice, knowledge, and mastery. New challenges cannot be solved without improving on old knowledge.	GD35
	Fish tanks	Increase the level of complexity.	GD36
	Information just in time and on demand	Provide a hint to the player exactly when it is needed.	GD37

Pleasantly frustrating	The player faces problems that are a bit out of their comfort zone to enter the zone of proximal development and enter a state of flow.	GD38
Sandboxes	Bounded, protected spaces the player can explore.	GD39
Skills as strategies	Provide contexts to skills that the player needs to learn (e.g., avoid kill and drill).	GD40
Well-ordered problems	Engaging players in solving problems of increasing difficulty.	GD41

Grover and Pea (2013) also recently completed a comprehensive literature review examining the state of the field of CT, finding that no one definition had been agreed upon, summarizing tools that cultivate CT skills and the components of CT.

For the purpose of in-class assessment, Brennan and Resnick (2012) qualified elements of CT into various categories: computational concepts, computational practices, and computational perspectives. Brennan and Resnick use examples from children ages 8-17; the participants in this paper were pre-service teachers and the perspective uses gaming artifacts to focus on practices and perspectives of CT.

Basawapatna, Koh, Repenning, Webb, and Marshall (2011) created a "visual semantic evaluation tool" (Koh, Basawapatna, Bennet, & Repenning, 2010, p. 59) in order to qualitatively contrast and compare CT patterns in games. This tool was primarily used to compare student's games to the corresponding tutorials.

The qualitative properties that comprise good GD and CT have been defined, but there are no established tools for quantitatively evaluating GD and the CT skills associated with this process. The purpose of the survey instrument explored in this paper, the *Quality Practices of Game Design Survey* (*QPGDS*), is to provide a quantitative informative tool measuring the highlighted areas of CT and GD for games created in *Scratch* (MIT, 2009).

Quality Practices of Game Design Survey (QPGDS)

Tool Development

The survey tool was developed by two educators with a background in CS and Education. After completing a literature review on CT, including the components of CT sample games were examined. Specialists in the area of assessment, educational psychology, and video gamers were consulted in the finalizing of the tool. A sample of 40 games was proportionally and purposefully sampled from a pool of 166 games across different genres.

The Survey

The *QPGDS* is divided into seven major categories grouped by shared concepts: Problem Solving Opportunities, Computational Thinking, Customization of Player Experience, Player Immersion, Player Interaction, Player Motivation, and Interface Usability. What follows is an explanation of the seven categories that were derived from an extensive review of the literature on CT skills and of quality GD practices.

The QPGDS is meant to be a comprehensive comparison tool that can compare students' games in different dimensions of GD and CT. For example, after using our tool, two games could be compared and one could be identified as providing higher player motivation or containing higher-level understanding of CT.

The items in the *Quality Practices of Game Design Survey* (*see Table* 2.4*)* are mapped to the components in Table 2.2 and 2.3 using the coding system CT## and GD## respectively.

Table 2.4
The Quality Practices of Game Design Survey (QPGDS).

Problem Solving Opportunities	
Game contains puzzles (0-2) CT34, CT36, GD03, GD15	
0 Points	No puzzles present.
1 Point	Offers simple puzzles. I.e. remembering, understanding, or applying.
2 Points	Has complex puzzles requiring effort to solve. I.e. analyzing, evaluating, or creating.
Forgiveness (0-1) GD08, GD20, GD39	
0 Points	Cost of failure is high. E.g., no checkpoints or lives system are implemented; player must begin game from the beginning.
1 Point	Cost of failure is low. I.e., some system exists so the player does not need to start from the beginning. E.g., lives or checkpoints.
Goals (0-1) GD03, GD05, GD06, GD15	
0 Points	There are no goals or objectives present in the game.
1 Point	There are goals or objectives present in the game.
Challenge incrementally increases as the game progresses (0-1) GD03, GD10, GD18, GD24, GD27, GD35, GD38, GD41	

E.g., enemies get faster or stronger as the game progresses.	
0 Points	Levels do not get harder, enemies do not get faster or stronger, etc. as the game progresses.
1 Point	Levels get harder, enemies get faster or stronger, etc. as the game progresses.
Total Points (/5)	

Computational Thinking

Modularization of game elements to a specific object (0-4) CT10, CT11, CT12, CT14, CT15, CT18, CT22, CT23, CT25, CT29, CT37, CT39, CT40, CT41, CT42, CT43, CT44, CT50, CT52, CT53, CT56, CT73, CT77, CT83, CT86, CT89, CT90, CT95 E.g., sounds, variables and code applying to a specific sprite appear there.	
0 Points	All of sound, variables and code elements pertaining to an object appear on different objects (if they are present).
1 Point	One of sound, variables and code elements pertaining to an object appear on the appropriate objects (if they are present).
2 Points	Two of sound, variables and code elements pertaining to an object appear on the appropriate objects (if they are present).
3 Points	Sound, variables and code elements pertaining to an object appear on the appropriate objects (if they are present). Variable operations occur on appropriate objects.
4 Points	Sound, variables and code elements pertaining to an object appear on the appropriate objects and data is encapsulated on an object. Variables are appropriately localized or globalized.
Effective use of procedures/methods (0-2) CT01, CT11, CT12, CT14, CT18, CT19, CT29, CT30, CT37, CT39, CT40, CT41, CT42, CT43, CT44, CT50, CT52, CT53, CT57, CT73, CT77, CT78, CT86 I.e. use of procedures or methods.	
0 Points	No subroutines, procedures or methods exist within an object.
1 Point	Some subroutines, procedures or methods exist within an object, but additional code segments could optimally be refactored into subroutines, procedures or methods.
2 Points	All code segments that could be optimally be refactored into subroutines, procedures or methods are represented as such.
Recursive/iterative vs. linear structures (0-2) CT01, CT02, CT07, CT10, CT11, CT12, CT14, CT18, CT19, CT21, CT23, CT24, CT28, CT29, CT30, CT39, CT40, CT41, CT44, CT45, CT52, CT53, CT57, CT66, CT77, CT78, CT82, CT86, CT94	
0 Points	Iterative structures are not used. E.g., CHANGE TO THIS COSTUME. CHANGE TO THIS COSTUME instead of NEXT COSTUME in a loop.
1 Point	Some linear structures are present where recursive or iterative structures could be used.
2 Points	Wherever possible iterative or structures are used in place of linear methods. Redundancies are removed. E.g., recursion or loops.
Cascading changes driven from one location (0-2) CT11, CT29, CT36, CT39, CT44, CT52, CT53, CT65, CT78, CT82, CT86 I.e. multiple locations are not sending events (space bar starts event and a variable or broadcast passes the message vs every location separately sensing the space bar press). E.g., a variable could drive all changes.	
0 Points	No changes are driven by the same events where appropriate.
1 Point	Some changes are driven by the same events where appropriate.
2 Points	All changes cascading in game are sent from one event where appropriate.
Accurate use of mathematical modeling (0-2) CT08, CT11, CT15, CT24, CT29, CT33, CT37, CT39, CT44, CT45, CT51, CT52, CT53, CT56, CT77, CT86 E.g., such as acceleration or gravity. E.g., if the spacebar is pressed to jump instead of increasing y, then decreasing y, the game has some sort of velocity or variable modeling.	

0 Points	No mathematical models are present or models are not effectively demonstrated.
1 Point	Mathematical models are present, but the game does not use effective models to simulate them.
2 Points	Mathematical models are present and the game uses effective models to simulate them.
Proper state initialization (0-1) CT11, CT13, CT29, CT39, CT44, CT52, CT53, CT64, CT86, CT97, , CT98 E.g., variables, costumes, sounds, actor locations.	
0 Points	Game state is not properly initialized. Some states are persistent from different game runs causing errors
1 Point	State is correctly initialized where required.
Game play is clean with at most minor programming errors (0-1) CT05, CT11, CT20, CT29, CT35, CT39, CT44, CT52, CT53, CT86	
0 Points	Game play suffers greatly from programming error
1 Point	Game play is not hampered by programming error.
Events are driven smoothly in parallel (0-2) CT05, CT11, CT29, CT35, CT36, CT39, CT44, CT49, CT52, CT53, CT65, CT68, CT78, CT82, CT86, CT96	
0 Points	No parallel processes exist throughout game.
1 Points	Some states carry into other states as code is not set to work concurrently with other code.
2 Point	No noticeable errors; code successfully works in parallel.
Total Points (/16)	

Customization of Player Experience

Multiple playthroughs yield different experiences (0-1) CT15, GD04, GD07, GD09, GD19, GD31, GD34, GD39 I.e. Exploring is an option	
0 Points	Multiple playthroughs yield the same experience.
1 Point	Multiple playthroughs yield different experiences.
Allows multiple ways through the game based on player choices (0-2) CT15, GD04, GD07, GD09 GD13, GD19, GD31, GD34	
0 Points	Player is not given opportunities to make choices throughout the game. I.e., linear game.
1 Point	Player can make minor choices affecting game. I.e., choosing a path right or left path. I.e., picking up different items throughout the game.
2 Points	2 Points - Player is free to make a wide range of choices. I.e., synthesizing a variety of items in Minecraft.
Players can customize the PC[1] (0-1) CT15, GD04, GD09, GD12, GD19, GD31, GD32, GD33	
0 Points	The PC is not customizable.
1 Point	The PC can be customized.
Total Points (/4)	

Player Interaction

NPC[2] interactions with other NPCs (0-1) CT15, GD01	
0 Points	No NPC interactions with other NPCs.
1 Point	NPC interactions with other NPCs.
Player interactions with NPCs (0-3) CT08, CT15, GD01, GD34	
0 Points	No PC interactions with NPCs.
1 Point	Interactions with NPCs are linear and scripted throughout the game. E.g., NPCs may repeat the same line over and over again.
2 Points	Interactions with NPCs change throughout the game. E.g., can change throughout story completion. E.g., can randomly be selected to display at a time.
3 Points	PC actions affect NPCs reactions to the PC.

	E.g., PC kills a family member of the NPC and can no longer interact friendly with that character. E.g., PC completes a quest for the NPC making the NPC trust the PC with further tasks.
Information is revealed as the player needs it (0-1) CT08, GD16, GD37	
0 Points	No information is revealed to the player.
1 Point	Information is told explicitly to the player. E.g., in *God of War*, the player is told that they need to press certain buttons at certain times during reaction, quick-time events.
2 Points	Information is shown implicitly to the player. E.g., in *Mega Man*, some obstacles or enemy behaviours are shown to the player before they encounter them so they can deduce a proper action to take when they encounter that obstacle or enemy.
A reference can be looked at if the user desires (0-2) CT08, GD04, GD16, GD37 E.g., Having the control list accessible when the player presses the 'h' key.	
0 Points	No help reference is available.
1 Point	Player controls or other hints are shown sometime in the game, but are not accessible over the course of the game. E.g., Controls are displayed at the beginning
2 Points	Player controls or other hints are accessible over the course of the game.
Total Points (/7)	

Player Immersion

Characters' animation (0-2) CT15, CT51	
0 Points	Characters are not animated.
1 Point	PC is animated based on player actions (internal). E.g., when buttons are pressed, player's death (reaction).
2 Points	Environment and NPCs are animated (external) E.g., NPC's reaction to player's attacks.
Music, sound, and animation (0-3) CT15, CT51	
0 Points	Music, sound, and animation are absent from the game.
1 Point	One of Music, sound XOR animation is extremely well done.
2 Points	Two of music, sound, and animation are unified.
3 Points	Music, sound, animation, and tasks all create a unified experience.
Total Points (/5)	

Player Motivation

Player is motivated to complete tasks given to PC (0-3) GD02, GD07	
0 Points	No high score functionality or storyline.
1 Point	High score functionality included.
2 Points	Storyline included.
3 Points	Storyline is developed throughout the game and has an ending.
Users assume a role in the game, rather than simply playing (0-2) GD09, GD14, GD33	
0 Points	Players do not take on or create a persona that they develop throughout the game.
1 Point	Players take on or create a persona that is not developed throughout the game.
2 Point	Players take on or create a persona that is developed throughout the game.
Achievements present in the game (0-2) GD02	
0 Points	No immediate, meaningful feedback is given to the player upon accomplishing optional tasks in the game.
1 Point	Achievements are present, but the player isn't alerted when these are accomplished.
2 Points	Immediate, meaningful feedback is given to the player upon accomplishing additional tasks in the game.
Rules, goals and objectives are explicit (0-1) GD05, GD06	
0 Points	The rules, goals, and objectives the player must complete are NOT explicit; the player does NOT know what they must accomplish.
1 Point	The rules, goals, and objectives the player must complete are explicit; the player knows what they must accomplish.

Total Points (/8)	

Interface Usability	
Meaningful feedback (0-1) CT08 E.g., when buttons are clicked, there is visual or auditory feedback.	
0 Points	The game does not provide feedback to the player when there is a change of state.
1 Point	The game provides feedback to the player when there is a change of state.
Uses traditional control conventions (0-1) GD21, GD29 If the game warrants using an existing control schema, that is the one that is used. E.g., movement using ASDW or up, down, left, right.	
0 Points	When appropriate to use an existing control scheme, this is ignored and a traditional game control convention is not used.
1 Point	If appropriate, traditional game control conventions are used.
Affordance and Visibility (0-1) CT08, GD21, GD19, GD31	
0 Points	Objects that the player needs to interact with are NOT salient or it is not obvious with what or how the player should interact with objects.
1 Point	Objects the player needs to interact with are salient and it is obvious with what or how the player should interact with objects.
Total Points (/3)	

Notes. [1] PC refers to the Player Character in a game.
[2] NPC refers to the Non-Player Character(s) in a game.

Rationale for Instrument Development

Problem Solving Opportunities (0-5). Problem Solving Opportunities have to do with the puzzles that the designer has created for the player to contend with. These are important to engaging the player through an appropriate level of challenge.

Game contains puzzles (0-2). Puzzles that require effort to solve provide opportunities for additional cognitive processing (Rice, 2007). We have chosen to split this section of the survey up into higher- and lower-end problems. Higher-end problems use the higher sections of Bloom's taxonomy: analyzing, evaluating or creating (e.g., synthesizing a new item in *Minecraft* to complete a goal). Lower-end problems use: remembering, understanding, or applying (e.g., recalling and applying a certain sequence of buttons to perform a learned counter move in *Batman: Arkham City*).

Forgiveness (0-1). Gee (2003) identifies video games as their own literacy that supports learning. One learning theory that most games have deeply ingrained in them is a low cost of failure, which is usually found in the form of lives, checkpoints, or continues. If players do not need to restart the game from the beginning to retry where they made an error, they can focus on the area where they had a problem. Gee (2013) also expresses that players may be more willing to make errors and explore, if the cost of failure is low. The "consequences of failure" (Gee, 2005, pg. 35) are lowered and risk taking is encouraged.

Goals (0-1). Games require objectives for the player to complete or else the 'game' may be more of a simulation than an actual game.

Challenge incrementally increases as the game progresses (0-1). Well-ordered, pleasantly frustrating problems should prepare players for challenges that they encounter later in the game: "early problems set up later success" (Gee, 2013; e.g., enemies or puzzles become more difficult as the game progresses). Video games should be built on traditional educational scaffolding to help the player learn new skills throughout the game. Good games integrate the cycle of expertise requiring players to build upon their previous knowledge to complete new challenges (Gee, 2005).

Computational Thinking (0-16). There has been a lot of work on identifying and qualifying the various components of CT. This section aims to provide grounds for measuring CT. As a video game is the artifact that is being analyzed, the students' programming, and how the overall game, is how the students CT competency is being assessed. "Programming is not only a fundamental skill of CS and a key tool for supporting the cognitive tasks involved in CT but a demonstration of computational competencies as well" (Grover & Pea, 2013, p. 40). This section of the rubric concerns itself with optimal algorithms for the game and being easily decipherable by a human looking over the game code.

Modularization of game elements to a specific object (0-4). Abstraction, or the hiding of details of an object, can help encapsulate game objects into logical and functional entities that function independently and only interact at a higher level. Wing (2008) aptly calls this "hierarchical decomposition" (p. 3720) as components should be separated by functional capacity in a hierarchy. The main components that were found in our dataset require being modularized to particular objects include: sound, variables, and scripts. In *Scratch* (MIT, 2009), variables can be set up to be either local, to an object or global. To receive full points in this area, variables must be local or global where appropriate.

Effective use of procedures/methods/functions (0-2). Code on an object can be further modularized into procedures or methods. If one script on an object is ran frequently, this script could be separated and refactored as a separate function that can be called. For full points in this area, all segments of code that could be refactored into a subroutine are represented as such.

Recursive/iterative vs. linear structures (0-2). This category has to do with coding style and abstract understanding of computational theory. The lowest category would be for code that completely appears in a linear structure. This is not feasible as code quickly becomes redundant, and involves a lot of vertical scrolling, thereby making it difficult for humans to read. Through the use of recursive or iterative structures, code is cleaner.

Cascading changes driven from one location (0-2). This category has to do with removing redundant state changes, encapsulating data correctly and having an event driven game that functions properly with parallel scripts running. Game state changes are only driven from one location (e.g., multiple locations should not be sending events; i.e., space bar starts event and a variable or broadcast passes the message versus every location separately sensing the space bar press).

Accurate use of mathematical modeling (0-2). If any interactions utilizing physics occur, these are accurately modeled in the game using appropriate velocity and acceleration laws (i.e., if a character jumps in a platformer game, there is a force that pushes the character back down; e.g., gravity).

Proper state initialization (0-1). There are many states that must occur over the course of a game's lifetime, but what makes them effective is having them be in the proper state at the proper time. This category has to do with initializing all components to the proper state at the beginning of the game (e.g., variables, costumes, sounds, actor locations).

Game play is clean with at most minor programming errors (0-1). Game play is not hampered by programming error. In commercial games, this is usually not a problem, but in student games this occasionally can be one. Perlis (1962) discusses "priori attention to eventualities regardless of likelihood" (p. 192) discussing how even the most infrequent cases need to be accounted for in the game's algorithms.

Events are driven smoothly in parallel (0-2). Games are all run by events or interactions with the player. There are usually multiple events running at a time and a game needs to be able to handle these in the correct order.

Customization of Player Experience (0-4). Constructionist theory hinges on learners creating an artifact representing their understanding. Video games are quite similar in that the artifact that gamers are creating is their play through a game. Having a customized game experience creates immersion and a unique personalized experience.

Multiple playthroughs yield different experiences (0-1). Multiple playthroughs do not yield the same linear experience. Gee (2013) discusses how the best games allow the player to explore and experiment with different styles and role play scenarios that they would not traditionally get to participate in.

Allows multiple ways through the game based on player choices (0-2). This can mean multiple paths or choices to get the player through the game. This falls under Gee's (2013) Co-design and Manipulation principles. The player should have control over their journey through the game whether by making their playthrough unique to them or by being able to manipulate a wealth of variables within the game.

Players can customize the Player Character (0-1) . The game allows the player to customize the player in some way. Effectively developed characters entrance players in the game (Gee, 2005). Role Playing Games (RPGs) are excellent for this criterion as the players frequently get a large say in how a character is customized. This also falls under Gee's (2013) co-design principle, what the player does matters and changes the game experience.

Player Interaction (0-7). Games are meant to be an interactive, dynamic experience that the player takes part in as opposed to a passive experience. The more interactive a game is, the more involved the player will be in the game.

NPC interactions with other NPCs (0-1). In a fully immersive world, NPCs do not just interact with the player. They also react and have relationships with one another. This is shown in a quality video game.

Player interactions with NPCs (0-3). The player has more buy-in when interacting with others in a game. The more realistic these characters are, the more buy-in the player will have. This can be made more realistic if NPCs do not just say the same lines of dialogue repeatedly. If NPCs react to what is going on in the world around them and the actions the player has taken, the world becomes more believable.

Information is revealed as the player needs it (0-2). Gee (2013) discusses information being available just in time (i.e. telling a hint exactly when it is needed; e.g., in *Mega Man 8*, there is a level where a platform drops very quickly; right before the player has the opportunity to jump on one, the game allows the player to see a platform drop in a similar fashion showing what is about to come). These two examples demonstrate explicitly (telling) versus implicitly (showing) revealing information to the player.

A reference can be looked at if the user desires (0-2). Gee (2013) discusses information being available on demand, which allows players to look up information when they want to know something. In *Ni no Kuni: Wrath of the White Witch*, there is a complete compendium the player can open to provide an overview of spells they have learned, a complete bestiary of enemies, world lore, as well as much more. Also, different learning styles may prefer to read information than have it told by a NPC.

Player Immersion (0-5). While graphics may not be required for a game to be good, they can contribute to a polished product which can engross the player.

Characters animation (0-2). Examining the highest rated games of all time, all include animated characters (Metacritic, n.d.). This category is tiered on our rubric as students in the course traditionally do not animate all states of their characters.

Music, sound, and animation (0-3). Extra Credits (2014a), a YouTube video game development and culture analytical channel, discuss the Magic Circle: a zone that the audience/player must cross to be enthralled in a medium/game. In this video, they state that "high value is placed on maintaining an emotional baseline" in society and in movies, video games and other media, music is the prompt to let a viewer know that it is okay to feel. Music, sound, animation help create the necessary buy-in to engage gamers. This category is tiered as students may not necessarily have all the components to make their game fully immersive.

Player Motivation (0-8). Why should a player even buy into a game? Motivating the player is an important part to creating a game. If a designer wants a player to continue playing their game, engaging incentives must be provided. This can be done many different ways.

Player is motivated to complete tasks given to PC (0-3). While score can be a motivator to play games, narrative is a far better motivator. With the exception of the yearly released sports games, all of the highest rated games of all time have some element of narrative developed through the game (Metacritic, n.d.). In addition, Rice (2007) emphasizes the story in his Video Game Higher Order Thinking Evaluation Rubric and does not even discuss score; stories with narrative provide more of a challenge for the player. Gee (2006) writes that "[h]umans find story elements profoundly meaningful and are at a loss when they cannot see the world in terms of such elements" (p. 2). We are therefore assuming that games with a narrative are more engaging than ones with a score.

Users assume a role in the game, rather than simply playing (0-2). Instead of dropping the player into a game as an anonymous shooter or vehicle, the player must don a role. "Users will engage in additional cognitive processing when role play is involved because it forces them to process information outside their normal experiences" (Rice, 2007, p. 94). Gee (2013) writes that good games should involve both body and mind, as users become engaged in their role.

Achievements present in the game (0-2). One psychological mechanism that many Massive Multiplayer Online Role Playing Games (MMORPGs) are strongly built on is the mechanism of achievements. Achievements recognize players for optional challenges they accomplish in-game. As soon as players finish an optional goal (usually requiring additional effort), they can immediately be rewarded and recognized with accolades, trophies, achievements, etc. These further incentivize the player to keep playing and achieve more. Dickey (2005) calls these hooks affirmations of performance.

Rules, goals and objectives are explicit (0-1). In order to differentiate a game and other interactive visual media, a video game must have rules, goals, and objectives. In order to avoid player frustration, these should be explicit.

Interface Usability (0-3). Video games are a type of graphical user interface (GUI) and therefore should follow some of the standards that GUIs do. Don Norman (2002) has established a set of design principles that are taught in software engineering that are upheld in quality interfaces.

Meaningful feedback (0-1). The game should provide meaningful feedback to the player. Feedback consists of giving the user information about what action has been completed and allows the user to continue on with this information (Norman, 2002; e.g., when an enemy ship is hit, it could blink to indicate that there has been a hit, and let the player know that damage has been taken. Alternatively, if the player can choose to arm three different weapons, perhaps the loaded weapon glows on the screen to indicate it is selected).

Uses traditional control conventions (0-1). If an existing control scheme exists, that is the one that is used. This falls under Norman's (2002) Mapping and Consistency principles. Mapping is the motivation for the original usage of the arrow keys in games. It deals with having a link between control and effect (e.g., using the up arrow moving the character up). Consistency deals with using conventions if they already exist allowing for cognitive transference (e.g., the PC can be moved by using ASDW or up, down, left, right).

Affordance and visibility (0-1). Objects are salient in a way that the player knows they should interact with it. Visibility concerns itself with how likely the user is to know what to do next and Affordance is what characteristics, often physical, give a clue on how to use it (Norman, 2002). Extra Credits (2014b) discuss affordances and clearly outline affordances saying "form implies function;" if there are rectangular and circular objects near a cash register, it can be inferred that these might be bills and coins.

These areas have been identified as constructs beneficial to pre-service teachers. Every experience that teachers design for a player in a game can be transferred to a student in a classroom. Problem solving opportunities provide challenges to the player and the effort invested creating quality opportunities can be applied to challenges presented to students in class. CT is a systematic problem solving method that can be inferred through the investigated digital game artifacts, which can be passed on to future students to use in their careers. Customizing the Player Experience, similar to constructionist theory, allows the player a personalized experience. Customizing a student's experience can allow them to articulate or understand material in a way that is meaningful for them. Player Interaction measures how interactive the game is verifying that there are enough interactive elements in the game to engage the player; students' classroom experiences should be active and engaging. Player Immersion focuses on creating a unified environment for the players. While these are aesthetic, they allow students to buy-in to a more realistic environment. Player Motivation looks for at the incentive players have to continue to play the game or comparatively for students to continue to be engaged in class material. Interface Usability measures the ease-of-use for the interface and game elements, which is important as the tools teachers give their students should be intuitive to use. In the next section of this paper, the setting of this study is highlighted.

Methods

Participants

The pre-service K-12 teachers involved in this study chose EDIT 486, the course outlined in this paper, as an elective at a large Western Canadian university. Students had the choice whether or not to provide consent to participate in this study. A large majority of the students provided consent. In 15 course sections over a 48 month period, we had 166 participants: 60 (36%) females and 106 (64%) males. While this sample size was relatively small, the group is representative of diverse demographics and academic focuses within Education.

Context

EDIT 486 is a senior undergrad, educational technology elective course taught at the University of Alberta through the Educational Psychology department. The first iteration of the course was outlined by Boechler, Carbonaro, deJong, Stroulia & Gutiérrez (2011). In this course, pre-service teachers become builders of video games showcasing constructionism in practice. By the end of the course pre-service teachers should be able to use video games in their classrooms and implement the acquired pedagogy. Students use the graphic programming environments *Scratch* (MIT, 2009) and *Kodu* (Microsoft, 2014), utilizing one environment for the first half of the semester and the other for the second half.

The course has a 50/50 lecture/lab format running twice a week in three hour periods. The lab period is further divided into two components: for the first half a teaching assistant, who is well-versed in *Scratch* (MIT, 2009) and *Kodu* (Microsoft, 2014), walks students through essential techniques for GD. The second half of this time period is ‘open lab’ time for students work on assigned tutorials derived from the textbook *Scratch for Teens* (Ford, 2008). Students are required to complete three of these short tutorials a week where they learn the basics of the interface. In addition, students are encouraged to personalize the assignments to better their understanding of the interface, as well as to explore potential ideas for their *Scratch* (MIT, 2009) video game project. For the duration of the ‘open lab’ time the teaching assistant is on hand to answer any questions students may have.

After students complete all the tutorials they must make a *Scratch* (MIT, 2009) video game project on a topic of their choice. These *Scratch* (MIT, 2009) projects are examined in this paper. As the course focuses on constructionism, students are encouraged to build their knowledge of a topic into the game as opposed to directly making an educational game. For example, a student’s understanding of gravity could be developed into a game where gravity is constantly acting on a game character as opposed to using a quiz or kill and drill game expressing the physics formulas used to represent the force of gravity. Students receive two dedicated weeks to complete their projects after their tutorials are complete, but can start working on their projects at the start of the course. After completing the project, students write a personal reflection on the GD process concentrating on their areas of learning from their experience creating the game.

The lecture covers various types of constructivism, constructionism, technology taxonomies, video game literacies, integrating video games into the classroom, programming basics, applications in education, among others. Students have weekly assigned readings that they take turns presenting on in groups. These presentations include an online discussion, an in-class discussion, as well as an activity that utilizes a technology to go over the main points of the readings. The final paper for this course allows the students to create a plan on how they would implement *Scratch* (MIT, 2009) or *Kodu* (Microsoft, 2014) in a classroom in their choice of subject area; they must support their decisions with literature.

Genre Category Classification

A classification scheme was constructed to evaluate the games based on genre. No one schema has been universally agreed upon. Wolf (2008) identified over forty different video game genres, which communicates the complexity of clarifying genres. “The idea of genre has not been without difficulties, such as the defining of what exactly constitutes a genre, overlaps between genres, and the fact that genres are always in flux as long as new works are being produced (p. 1). Henry (2011) wrote his dissertation on the evolution of genre and put forward that most authors and publications use some form of the genres: Action, Adventure, Driving/Racing, Puzzle, Simulation, Sports and Strategy. Table 2.5 shows the genres this study identifies through a comprehensive literature review as well as being present in our dataset (Adams, 2007; AllGame, 2010; Apperley, 2006; Elliott, Ream, McGinsky, & Dunlap, 2012; ESA, 2013; GameFAQs, 2014; GameFly, Inc., 2003, February 28; Hamlen, 2011; Henry, 2011; IGN, 2014; Lee, Ko, Song, Kwon, Lee, Nam & Jung, 2007; Lucas & Sherry, 2004; Rollings & Adams, 2003; Ventura, Shute & Jeon, 2012; Whitton, 2014;Wolf, 2008).

Table 2.5

Genre classification scheme in our study with examples, findings and sources.

Genre	Summary	Observed Qualities
Action	This type of game focuses on reactions and timing.	
Action - Violent	Focus on violence as a major game mechanic.	● Score points ● Attempt to get high score ● Reaction tests ● Space and zombie themes
Action - Non-Violent	Action game with non-violent motif.	● Commonly collecting objects ● Commonly avoiding other characters or objects ● Quick-time events requiring quick reactions ● Getting points ● Trying to achieve a high score ● Platformers ● Frogger-type games ● Pong and Brickbreaker-type games
Adventure	This type of game focuses on player interactions based on developing the narrative.	● Narratives ● Fantasy-themed games - dragons, kings, swords and princess
Puzzle	This type of game focuses on logical reasoning and problem solving.	
Artificial Life	Modeling biological processes.	● Animals - notably reptiles (turtle and gecko)
(Construction And Management) Simulation	This type of game focuses on building a structure while managing resources.	

Role Playing	This type of game focuses on narratives emphasizing character development.	
Sports	This type of game focuses on athletic competition.	Only two Sports games were made.
Strategy	Resource management-style game play.	Only one strategy game was made.
Driving/Racing/ Vehicle Simulation	This type of game focuses on operating vehicles.	Only one Driving/Racing/Vehicle Simulation game was made.
Other		● Some educational games - label body parts or geographical areas ● Song creation soundboards

GameFAQs (2014) and IGN (2014) are two gaming website communities commonly used by gamers. In 1995, GameFAQs was founded and primarily contained cheat codes, frequently asked questions and walkthroughs for video games. IGN, the Imagine Games Network, was created in 1996 and primarily reviews commercial games before they are released. Both websites were also consulted in defining our schema.

Action. Action games are driven by the manipulation of onscreen elements (Allgame, 2010). Elements common to action games include: lives, quick-time events (reaction tests) and hand-eye coordination tests; often these games include levels, waves and power-up design elements (Rollings & Adams, 2003). Goals may be easy to perceive, but difficult to accomplish (Adams, 2007). Action games commonly also have checkpoints or enemy spawn points (Adams, 2007). These games only display the information players need to know; whenever possible this is done through graphical indicators instead of numbers or text (Adams, 2007).

Rollings & Adams (2003) differentiate action games into two categories: those with shooting (Action Shooter) and those without (Action Non-Shooter). This ‘shooter’ distinction can be somewhat misleading as the categories more so refer to the violent content in the games. Games containing violence, usually with weapons, fit into the Action Shooter category, whereas non-shooter action games have a non-violent theme. We have relabeled these categories to reflect the more inclusive definition that Rollings & Adams (2003) applied: Action Violent and Action Non-Violent.

Action violent. Violent action games tend to have two main characteristics: hit the enemy and avoid being hit. For example, in a Shooter game, the player must shoot the enemies, while refraining from being shot (Allgame, 2010). This type of game play has been called "kill-or-be-killed" (Elliott, Ream, McGinsky, & Dunlap, 2012, p. 954). Subgenres of video games that would fit in this category include: Shooters, Fighting and Violent Platformers. Commercially Off-the-Shelf (COTS) available examples of this type of game include:

- *Call of Duty: Advanced Warfare*
- *Far Cry 4*
- *Assassin's Creed Unity*

Action non-violent. These are action-based games requiring quick reactions or coordination, but do not emphasize combat of any form. A non-violent platform game would be one that involves dodging collisions with enemies while traversing an environment requiring precision timing of movement and jumping to reach an objective (Elliott, Ream, McGinsky, & Dunlap, 2012). COTS available examples of this type of game include:

- *LittleBigPlanet 3*
- *Donkey Kong*
- *Super Monkey Ball 2*

Adventure. Rollings & Adams (2003) identify an Adventure game as a game centered on a character that has an interactive story. Some Adventure games can take over 40 hours to complete the first time, but if the player knows how to solve the puzzles and what routes to take, this could be reduced to 4 hours (Adams, 2007 Wolf (2008) elaborates that these type of games are non-linear, and thereby allow the player to explore the game's environment. Action and other elements may be present in an Adventure game, but these elements are secondary to exploration and advancement of the narrative (Allgame, 2010). Common conceptual puzzles in these games include: finding keys to locked doors, obtaining inaccessible objects and collecting things (Adams, 2007). COTS available examples of this type of game include:

- *The Walking Dead*
- *Lego Batman 3*

Puzzle. Rollings & Adams (2003) define puzzle games as ones that primarily center on the act of puzzle solving; sometimes these puzzles are separate from a narrative. COTS successful puzzle games need to be challenging, visually attractive and enjoyable (Adams, 2007). Games usually involve matching and logical reasoning (Elliott, Ream, McGinsky, & Dunlap, 2012). COTS available examples of this type of game include:

- *Scribblenauts Unmasked: A DC Comics Adventure*
- *Pokémon Battle Trozei*
- *Tetris*

Artificial life. Artificial Life games model biological processes (Rollings & Adams, 2003). These digital creatures can die without proper care from the player (Wolf, 2008). COTS available examples of this type of game include:

- *Tamagotchi*
- *Spore*
- *The Sims*

Construction and management simulation. Construction and Management Simulation games are about balancing limited resources to build or expand a business, community or empire while contending with other internal variables (i.e. Crime; Wolf, 2008); the player can control building locations, factors to reduce prices, and also demolition (Rollings & Adams, 2003).

There is an emphasis on cause-and-effect (Allgame, 2010). COTS available examples of this type of game include:

- *Roller Coaster Tycoon*
- *SimCity*
- *FTL: Faster Than Light*

Role Playing. Role Playing games also contain strong storylines, but focus on player characters that improve through experience over the duration of the game; the in-depth storyline allows the player to sympathize with the main characters (Rollings & Adams, 2003). Commonly the main character completes minor quests or tasks contributing to the larger, central goal in the game (Allgame, 2010). Role playing games tend to have a mechanic involving the creation of a customized, powerful party (Elliott, Ream, McGinsky, & Dunlap, 2012). COTS available examples of this type of game include:

- *Diablo III*
- *Final Fantasy X*
- *South Park: The Stick of Truth*

Sports. Sports-type games are based on athletic competition (Rollings & Adams, 2003). COTS available examples of this type of game include:

- *Mario Golf World Tour*
- *FIFA 15*
- *Tony Hawk's Pro Skater HD*

Strategy. Strategy games focus on resource management in either a turn-based or real-time environment; commonly they have three primary factors: the theme, the presentation layer and multiple perspectives (Rollings & Adams, 2003). This style of game play emphasizes strategy instead of fast action and quick reflexes (Wolf, 2008). Player decisions have cascading, long-term effects (Allgame, 2010). COTS available examples of this type of game include:

- *Sid Meier's Civilization V*
- *Total War: Rome II*
- *Command & Conquer*

Driving/racing/vehicle simulation. Rollings & Adams (2003) identify a game as a Vehicle Simulation game if the player is immersed in a vivid driving/flying situation, real or imaginary; these vehicles can include: flight, driving, boats or ships, tanks or spacecrafts. Driving/Vehicle Simulation-type games focus more on creating a genuine driving experience, whereas this may come secondary in a Racing-type game as the primary goal is to get from one area to the finish line as quickly as possible. Wolf (2008) writes that Driving games focus on steering, manoeuvrability, speed control, and fuel conservation. Overall these genres are about operating a vehicle. COTS available examples of this type of game include:

- *Mario Kart 8*
- *Gran Turismo 6*
- *Blazing Angels*

Creation of the QPGDS

The survey tool was developed by two educators with a background in CS. They completed a literature review on CT, including the components of CT. The CT lit review can be seen in Table 2.2. The quality components of GD can be seen in Table 2.3. In addition, sample games were examined. Experts in the area of assessment, educational psychology, and video gamers were consulted in the finalizing of the tool.

Game Evaluation with the QPGDS

Forty was the decided number of games chosen to beta test the *QPGDS*. A purposeful subset was proportionally sampled from the overall pool of 166 games using the SPSS random selector (e.g., as Adventure games were 12% of the overall 166 created games, roughly 12% of the 40 games chosen were from the Adventure genre; *see Table* 2.*6*). Our pre-service teachers did not create any construction and management simulations. As the number of Action games was so large, they were further split up into Action Non-Violent and Action Violent.

Table 2.6
Frequency count for games of certain genres and the representative sample size.

Genre	Frequency	Percent (%)	Representative Sample (~40 games)
Action Non-Violent	69	41.6	16
Action Violent	35	21.1	7
Adventure	20	12	5
Artificial Life	3	1.8	1
Puzzle	23	13.9	5
Sports	2	1.2	1
Strategy	1	0.6	1
Driving	1	0.6	1
Miscellaneous	12	7.2	1
Total	166	100	40

After the subset was created, the 40 games were evaluated on the *QPGDS* (*see Table* 2.*4)*) using a Google Form. This evaluation consisted of playing the game through at least three times (more if the game had additional features) and then examining the game's code.

Data Collection

In order to anonymize the students' data, unique identification numbers were randomly assigned to the students. All their game data, demographic data is tied to their ID number and there is no biased pattern linking them to their gender, term that they took the course, or any other data collected. Two instructors jointly taught all the sections the course.

Data Analysis

The frequency tables showing how games scored on the *QPGDS* can all be found in the Appendix. The results are examined in the next section. Table 2.9 contains a Pearson correlation chart showing how criteria correlated with one another. Tables 2.8 and 2.9 provide Pearson Correlations and *p* value colour schemes respectively. Very strong and strong relationships across criteria are discussed in the following section. Correlations could be subject to using such a small dataset.

Table 2.7
Colour scheme for identifying Pearson Correlations in Table 2.9.

Pearson's r Correlation	Relationship
Very Strong	± .70 to 1
Strong	± .40 to .69
Moderate	± .30 to .39
Weak	± .20 to .29
No or Negligible	± .01 to .19

Table 2.8
Colour scheme for p value significance for Pearson Correlations between items in Table 2.9.

***p* value Significance**
Correlation is significant at
0.01 level (2-tailed)
0.05 level (2-tailed)
> 0.05 Not Significant

Table 2.9
Pearson correlations between QPGDS items. This table is split horizontally over four pages.

All values rounded to the nearest hundredth N = 40		Game contains puzzles (2)	Forgiveness (1)	Goals (1)	Challenge incrementally increases (1)	Modularization of game elements (4)	Procedures/methods (2)	Recursive or iterative structures (2)	Cascading changes (2)
Game contains puzzles (2)	Pearson Correlation		0.40	0.47	0.19	-0.08	0.37	-0.02	0.37
	Sig. (2-tailed)		0.01	0.00	0.25	0.64	0.02	0.89	0.02
Forgiveness (1)	Pearson Correlation	0.40		0.40	0.24	0.26	0.05	0.06	0.05
	Sig. (2-tailed)	0.01		0.01	0.14	0.11	0.74	0.72	0.74
Goals (1)	Pearson Correlation	0.47	0.40		0.19	-0.20	-0.07	-0.02	-0.07
	Sig. (2-tailed)	0.00	0.01		0.25	0.21	0.69	0.89	0.69
Challenge incrementally increases (1)	Pearson Correlation	0.19	0.24	0.19		0.22	0.04	0.08	0.04
	Sig. (2-tailed)	0.25	0.14	0.25		0.18	0.81	0.61	0.81
Modularization of game elements (4)	Pearson Correlation	-0.08	0.26	-0.20	0.22		-0.09	0.08	-0.09
	Sig. (2-tailed)	0.64	0.11	0.21	0.18		0.56	0.63	0.56
Procedures/methods (2)	Pearson Correlation	0.37	0.05	-0.07	0.04	-0.09		-0.12	1.00
	Sig. (2-tailed)	0.02	0.74	0.69	0.81	0.56		0.45	0.00
Recursive or iterative structures (2)	Pearson Correlation	-0.02	0.06	-0.02	0.08	0.08	-0.12		-0.12
	Sig. (2-tailed)	0.89	0.72	0.89	0.61	0.63	0.45		0.45
Cascading changes (2)	Pearson Correlation	0.37	0.05	-0.07	0.04	-0.09	1.00	-0.12	
	Sig. (2-tailed)	0.02	0.74	0.69	0.81	0.56	0.00	0.45	
Mathematical modeling (2)	Pearson Correlation	0.12	0.03	0.12	0.17	0.25	0.15	-0.01	0.15
	Sig. (2-tailed)	0.45	0.83	0.45	0.29	0.13	0.34	0.97	0.34
State initialization (1)	Pearson Correlation	-0.04	0.28	-0.04	-0.20	0.30	-0.05	-0.18	-0.05
	Sig. (2-tailed)	0.82	0.08	0.82	0.23	0.06	0.78	0.27	0.78
No programming errors (1)	Pearson Correlation	0.31	0.30	0.31	-0.04	-0.19	0.19	-0.07	0.19
	Sig. (2-tailed)	0.05	0.06	0.05	0.79	0.25	0.24	0.65	0.24
Parallel Events (2)	Pearson Correlation	-0.06	-0.03	-0.06	0.11	0.03	0.03	0.23	0.02
	Sig. (2-tailed)	0.72	0.86	0.72	0.52	0.86	0.88	0.15	0.88
Playthroughs yield different experiences (1)	Pearson Correlation	-0.06	0.21	-0.06	0.00	0.26	-0.17	-0.08	-0.17
	Sig. (2-tailed)	0.72	0.20	0.72	1.00	0.11	0.29	0.63	0.29
Multiple ways through game (2)	Pearson Correlation	-0.09	0.28	-0.09	-0.13	0.11	0.20	0.02	0.20
	Sig. (2-tailed)	0.60	0.08	0.60	0.42	0.49	0.22	0.92	0.22
Customizable PC (1)	Pearson Correlation	-0.26	-0.13	-0.26	0.15	0.13	0.11	0.11	0.11
	Sig. (2-tailed)	0.11	0.42	0.10	0.34	0.44	0.51	0.48	0.51
NPCs <> NPCs (1)	Pearson Correlation	0.07	-0.05	-0.37	-0.23	-0.01	0.08	-0.07	0.08
	Sig. (2-tailed)	0.69	0.74	0.02	0.15	0.95	0.62	0.68	0.62
PC <> NPC (3)	Pearson Correlation	0.09	-0.03	0.09	-0.09	-0.29	0.41	-0.12	0.41
	Sig. (2-tailed)	0.60	0.85	0.60	0.59	0.07	0.01	0.45	0.01
Information is revealed (1)	Pearson Correlation	0.10	0.25	0.10	-0.13	0.27	0.02	0.04	0.02
	Sig. (2-tailed)	0.54	0.12	0.54	0.41	0.10	0.90	0.79	0.90
Game reference (2)	Pearson Correlation	0.35	0.25	0.10	0.09	0.21	0.02	-0.07	0.02
	Sig. (2-tailed)	0.03	0.12	0.54	0.58	0.20	0.90	0.69	0.90
Characters animation (2)	Pearson Correlation	0.22	0.09	0.22	0.14	0.04	0.08	-0.06	0.08
	Sig. (2-tailed)	0.18	0.60	0.18	0.38	0.81	0.62	0.73	0.62
Music sound & animation (3)	Pearson Correlation	-0.07	0.05	-0.07	-0.16	0.01	-0.08	-0.32	-0.08
	Sig. (2-tailed)	0.69	0.74	0.69	0.34	0.95	0.61	0.05	0.62
Player is motivated do PC tasks(3)	Pearson Correlation	0.28	0.35	0.28	0.35	0.18	-0.04	-0.08	-0.04
	Sig. (2-tailed)	0.08	0.03	0.08	0.02	0.26	0.81	0.61	0.81
Player takes on role (2)	Pearson Correlation	-0.02	0.29	-0.02	0.29	0.30	0.07	0.19	0.07
	Sig. (2-tailed)	0.89	0.07	0.89	0.07	0.06	0.68	0.24	0.68
Achievements (2)	Pearson Correlation	0.08	0.00	0.08	0.07	-0.07	0.09	-0.13	0.09
	Sig. (2-tailed)	0.64	1.00	0.64	0.68	0.65	0.56	0.41	0.56
Explicit rules, goals, and objectives (1)	Pearson Correlation	0.10	0.25	0.10	-0.13	0.02	0.23	-0.29	0.23
	Sig. (2-tailed)	0.54	0.12	0.54	0.41	0.88	0.16	0.07	0.16
Meaningful feedback (1)	Pearson Correlation	0.61	0.66	0.61	0.15	0.13	0.18	-0.11	0.18
	Sig. (2-tailed)	0.00	0.00	0.00	0.34	0.44	0.27	0.48	0.27
Traditional control conventions (1)	Pearson Correlation	-0.04	0.28	0.70	0.13	-0.23	-0.05	-0.18	-0.05
	Sig. (2-tailed)	0.82	0.08	0.00	0.42	0.15	0.78	0.27	0.78
Affordance and Visibility (1)	Pearson Correlation	0.47	0.13	-0.05	0.19	0.18	0.37	-0.02	0.37
	Sig. (2-tailed)	0.00	0.42	0.75	0.25	0.27	0.02	0.89	0.02

All values rounded to the nearest hundredth	N = 40	Mathematical modeling (2)	State initialization (1)	No programming errors (1)	Parallel Events (2)	Playthroughs different experiences (1)	Multiple ways through game (2)	Customizable PC (1)	NPCs <> NPCs (1)
Game contains puzzles (2)	Pearson Correlation	0.12	-0.04	0.31	-0.06	-0.06	-0.09	-0.26	0.07
	Sig. (2-tailed)	0.45	0.82	0.05	0.72	0.72	0.60	0.11	0.69
Forgiveness (1)	Pearson Correlation	0.03	0.28	0.30	-0.03	0.21	0.28	-0.13	-0.05
	Sig. (2-tailed)	0.83	0.08	0.06	0.86	0.20	0.08	0.42	0.74
Goals (1)	Pearson Correlation	0.12	-0.04	0.31	-0.06	-0.06	-0.09	-0.26	-0.37
	Sig. (2-tailed)	0.45	0.82	0.05	0.72	0.72	0.60	0.10	0.02
Challenge incrementally increases (1)	Pearson Correlation	0.17	-0.20	-0.04	0.11	0.00	-0.13	0.15	-0.23
	Sig. (2-tailed)	0.29	0.23	0.79	0.52	1.00	0.42	0.34	0.15
Modularization of game elements (4)	Pearson Correlation	0.25	0.30	-0.19	0.03	0.26	0.11	0.13	-0.01
	Sig. (2-tailed)	0.13	0.06	0.25	0.86	0.11	0.49	0.44	0.95
Procedures/methods (2)	Pearson Correlation	0.15	-0.05	0.19	0.03	-0.17	0.20	0.11	0.08
	Sig. (2-tailed)	0.34	0.78	0.24	0.88	0.29	0.22	0.51	0.62
Recursive or iterative structures (2)	Pearson Correlation	-0.01	-0.18	-0.07	0.23	-0.08	0.02	0.11	-0.07
	Sig. (2-tailed)	0.97	0.27	0.65	0.15	0.63	0.92	0.48	0.68
Cascading changes (2)	Pearson Correlation	0.15	-0.05	0.19	0.02	-0.17	0.20	0.11	0.08
	Sig. (2-tailed)	0.34	0.78	0.24	0.88	0.29	0.22	0.51	0.62
Mathematical modeling (2)	Pearson Correlation		0.09	0.02	0.08	-0.05	0.14	0.52	0.07
	Sig. (2-tailed)		0.60	0.91	0.64	0.78	0.40	0.00	0.65
State initialization (1)	Pearson Correlation	0.09		0.22	-0.21	0.12	0.11	0.06	0.05
	Sig. (2-tailed)	0.60		0.18	0.20	0.45	0.49	0.71	0.78
No programming errors (1)	Pearson Correlation	0.02	0.22		-0.41	-0.08	0.17	-0.04	0.01
	Sig. (2-tailed)	0.91	0.18		0.01	0.62	0.28	0.81	0.95
Parallel Events (2)	Pearson Correlation	0.08	-0.21	-0.41		-0.07	-0.10	0.18	-0.02
	Sig. (2-tailed)	0.64	0.20	0.01		0.68	0.55	0.28	0.88
Playthroughs yield different experiences (1)	Pearson Correlation	-0.05	0.12	-0.08	-0.07		0.35	0.33	-0.22
	Sig. (2-tailed)	0.78	0.45	0.62	0.68		0.03	0.04	0.17
Multiple ways through game (2)	Pearson Correlation	0.14	0.11	0.17	-0.10	0.35		0.38	0.01
	Sig. (2-tailed)	0.40	0.49	0.28	0.55	0.03		0.01	0.98
Customizable PC (1)	Pearson Correlation	0.52	0.06	-0.04	0.18	0.33	0.38		-0.11
	Sig. (2-tailed)	0.00	0.71	0.81	0.28	0.04	0.01		0.51
NPCs <> NPCs (1)	Pearson Correlation	0.07	0.05	0.01	-0.02	-0.22	0.01	-0.11	
	Sig. (2-tailed)	0.65	0.78	0.95	0.88	0.17	0.98	0.51	
PC <> NPC (3)	Pearson Correlation	0.12	-0.11	-0.17	0.21	0.21	0.14	0.10	-0.01
	Sig. (2-tailed)	0.47	0.49	0.28	0.20	0.20	0.39	0.54	0.97
Information is revealed (1)	Pearson Correlation	0.09	0.24	-0.02	-0.17	-0.06	0.10	-0.25	-0.02
	Sig. (2-tailed)	0.58	0.13	0.89	0.30	0.73	0.52	0.12	0.90
Game reference (2)	Pearson Correlation	-0.04	0.24	-0.02	-0.06	-0.17	-0.13	-0.25	-0.02
	Sig. (2-tailed)	0.81	0.13	0.89	0.73	0.30	0.43	0.12	0.90
Characters animation (2)	Pearson Correlation	0.09	0.15	0.17	-0.01	-0.12	-0.02	-0.06	-0.08
	Sig. (2-tailed)	0.59	0.35	0.29	0.94	0.47	0.91	0.73	0.62
Music sound & animation (3)	Pearson Correlation	-0.07	0.56	0.19	-0.17	0.22	-0.01	0.11	0.08
	Sig. (2-tailed)	0.65	0.00	0.24	0.29	0.17	0.97	0.51	0.62
Player is motivated do PC tasks(3)	Pearson Correlation	-0.05	0.20	0.26	-0.11	0.21	-0.09	-0.15	-0.15
	Sig. (2-tailed)	0.76	0.23	0.11	0.52	0.19	0.59	0.34	0.34
Player takes on role (2)	Pearson Correlation	0.23	0.14	-0.07	0.23	0.13	0.02	0.27	-0.26
	Sig. (2-tailed)	0.14	0.37	0.65	0.15	0.42	0.92	0.10	0.11
Achievements (2)	Pearson Correlation	0.22	0.05	0.07	-0.09	-0.09	0.30	0.13	0.22
	Sig. (2-tailed)	0.17	0.74	0.67	0.60	0.60	0.06	0.44	0.17
Explicit rules, goals, and objectives (1)	Pearson Correlation	0.09	0.24	0.32	0.17	-0.17	-0.01	-0.08	-0.02
	Sig. (2-tailed)	0.57	0.13	0.04	0.30	0.30	0.94	0.61	0.90
Meaningful feedback (1)	Pearson Correlation	0.20	0.42	0.36	-0.02	0.14	0.10	-0.09	-0.18
	Sig. (2-tailed)	0.21	0.01	0.02	0.90	0.40	0.54	0.60	0.27
Traditional control conventions (1)	Pearson Correlation	0.09	-0.03	0.22	0.12	0.12	0.11	0.06	-0.56
	Sig. (2-tailed)	0.60	0.88	0.18	0.45	0.45	0.49	0.71	0.00
Affordance and Visibility (1)	Pearson Correlation	0.12	-0.04	0.31	-0.06	0.18	-0.09	0.09	0.07
	Sig. (2-tailed)	0.45	0.82	0.05	0.72	0.27	0.60	0.59	0.69

All values rounded to the nearest hundredth N = 40		PC <> NPC (3)	Information is revealed (1)	Game reference (2)	Characters animation (2)	Music sound & animation (3)	Player is motivated do PC tasks(3)	Player takes on role (2)	Achievements (2)
Game contains puzzles (2)	Pearson Correlation	0.09	0.10	0.35	0.22	-0.07	0.28	-0.02	0.08
	Sig. (2-tailed)	0.60	0.54	0.03	0.18	0.69	0.08	0.89	0.64
Forgiveness (1)	Pearson Correlation	-0.03	0.25	0.25	0.09	0.05	0.35	0.29	0.00
	Sig. (2-tailed)	0.85	0.12	0.12	0.60	0.74	0.03	0.07	1.00
Goals (1)	Pearson Correlation	0.09	0.10	0.10	0.22	-0.07	0.28	-0.02	0.08
	Sig. (2-tailed)	0.60	0.54	0.54	0.18	0.69	0.08	0.89	0.64
Challenge incrementally increases (1)	Pearson Correlation	-0.09	-0.13	0.09	0.14	-0.16	0.35	0.29	0.07
	Sig. (2-tailed)	0.59	0.41	0.58	0.38	0.34	0.02	0.07	0.68
Modularization of game elements (4)	Pearson Correlation	-0.29	0.27	0.21	0.04	0.01	0.18	0.30	-0.07
	Sig. (2-tailed)	0.07	0.10	0.20	0.81	0.95	0.26	0.06	0.65
Procedures/methods (2)	Pearson Correlation	0.41	0.02	0.02	0.08	-0.08	-0.04	0.07	0.09
	Sig. (2-tailed)	0.01	0.90	0.90	0.62	0.61	0.81	0.68	0.56
Recursive or iterative structures (2)	Pearson Correlation	-0.12	0.04	-0.07	-0.06	-0.32	-0.08	0.19	-0.13
	Sig. (2-tailed)	0.45	0.79	0.69	0.73	0.05	0.61	0.24	0.41
Cascading changes (2)	Pearson Correlation	0.41	0.02	0.02	0.08	-0.08	-0.04	0.07	0.09
	Sig. (2-tailed)	0.01	0.90	0.90	0.62	0.62	0.81	0.68	0.56
Mathematical modeling (2)	Pearson Correlation	0.12	0.09	-0.04	0.09	-0.07	-0.05	0.23	0.22
	Sig. (2-tailed)	0.47	0.58	0.81	0.59	0.65	0.76	0.14	0.17
State initialization (1)	Pearson Correlation	-0.11	0.24	0.24	0.15	0.56	0.20	0.14	0.05
	Sig. (2-tailed)	0.49	0.13	0.13	0.35	0.00	0.23	0.37	0.74
No programming errors (1)	Pearson Correlation	-0.17	-0.02	-0.02	0.17	0.19	0.26	-0.07	0.07
	Sig. (2-tailed)	0.28	0.89	0.89	0.29	0.24	0.11	0.65	0.67
Parallel Events (2)	Pearson Correlation	0.21	-0.17	-0.06	-0.01	-0.17	-0.11	0.23	-0.09
	Sig. (2-tailed)	0.20	0.30	0.73	0.94	0.29	0.52	0.15	0.60
Playthroughs yield different experiences (1)	Pearson Correlation	0.21	-0.06	-0.17	-0.12	0.22	0.21	0.13	-0.09
	Sig. (2-tailed)	0.20	0.73	0.30	0.47	0.17	0.19	0.42	0.60
Multiple ways through game (2)	Pearson Correlation	0.14	0.10	-0.13	-0.02	-0.01	-0.09	0.02	0.30
	Sig. (2-tailed)	0.39	0.52	0.43	0.91	0.97	0.59	0.92	0.06
Customizable PC (1)	Pearson Correlation	0.10	-0.25	-0.25	-0.06	0.11	-0.15	0.27	0.13
	Sig. (2-tailed)	0.54	0.12	0.12	0.73	0.51	0.34	0.10	0.44
NPCs <> NPCs (1)	Pearson Correlation	-0.01	-0.02	-0.02	-0.08	0.08	-0.15	-0.26	0.22
	Sig. (2-tailed)	0.97	0.90	0.90	0.62	0.62	0.34	0.11	0.17
PC <> NPC (3)	Pearson Correlation		-0.11	-0.10	0.13	0.00	-0.02	0.09	0.23
	Sig. (2-tailed)		0.52	0.52	0.44	0.98	0.89	0.58	0.15
Information is revealed (1)	Pearson Correlation	-0.11		0.52	0.19	0.02	0.24	0.15	0.04
	Sig. (2-tailed)	0.52		0.00	0.25	0.90	0.13	0.34	0.82
Game reference (2)	Pearson Correlation	-0.10	0.52		0.51	0.02	0.24	0.15	0.04
	Sig. (2-tailed)	0.52	0.00		0.00	0.90	0.13	0.34	0.82
Characters animation (2)	Pearson Correlation	0.13	0.19	0.51		0.27	0.37	0.35	0.18
	Sig. (2-tailed)	0.44	0.25	0.00		0.09	0.02	0.03	0.26
Music sound & animation (3)	Pearson Correlation	0.00	0.02	0.02	0.27		0.35	0.07	0.09
	Sig. (2-tailed)	0.98	0.90	0.90	0.09		0.03	0.69	0.56
Player is motivated do PC tasks(3)	Pearson Correlation	-0.02	0.24	0.24	0.37	0.35		0.33	0.10
	Sig. (2-tailed)	0.89	0.13	0.13	0.02	0.03		0.04	0.53
Player takes on role (2)	Pearson Correlation	0.09	0.15	0.15	0.35	0.07	0.33		0.03
	Sig. (2-tailed)	0.58	0.34	0.34	0.03	0.69	0.04		0.84
Achievements (2)	Pearson Correlation	0.23	0.04	0.04	0.18	0.09	0.10	0.03	
	Sig. (2-tailed)	0.15	0.82	0.82	0.26	0.56	0.53	0.84	
Explicit rules, goals, and objectives (1)	Pearson Correlation	0.01	0.17	-0.07	-0.03	0.23	0.24	0.26	0.04
	Sig. (2-tailed)	0.94	0.30	0.66	0.84	0.16	0.13	0.10	0.82
Meaningful feedback (1)	Pearson Correlation	0.22	0.08	0.25	0.21	0.18	0.31	0.19	0.13
	Sig. (2-tailed)	0.17	0.61	0.12	0.20	0.27	0.05	0.24	0.44
Traditional control conventions (1)	Pearson Correlation	0.23	-0.10	-0.10	0.15	-0.05	0.20	0.14	0.05
	Sig. (2-tailed)	0.15	0.52	0.52	0.35	0.78	0.23	0.37	0.74
Affordance and Visibility (1)	Pearson Correlation	0.09	-0.15	0.10	0.22	-0.07	0.28	0.21	0.08
	Sig. (2-tailed)	0.60	0.35	0.54	0.18	0.68	0.08	0.20	0.64

All values rounded to the nearest hundredth N = 40		Explicit rules, goals, and objectives (1)	Meaningful feedback (1)	Traditional control conventions (1)	Affordance and Visibility (1)
Game contains puzzles (2)	Pearson Correlation	0.10	0.61	-0.04	0.47
	Sig. (2-tailed)	0.54	0.00	0.82	0.00
Forgiveness (1)	Pearson Correlation	0.25	0.66	0.28	0.13
	Sig. (2-tailed)	0.12	0.00	0.08	0.42
Goals (1)	Pearson Correlation	0.10	0.61	0.70	-0.05
	Sig. (2-tailed)	0.54	0.00	0.00	0.75
Challenge incrementally increases (1)	Pearson Correlation	-0.13	0.15	0.13	0.19
	Sig. (2-tailed)	0.41	0.34	0.42	0.25
Modularization of game elements (4)	Pearson Correlation	0.02	0.13	-0.23	0.18
	Sig. (2-tailed)	0.88	0.44	0.15	0.27
Procedures/methods (2)	Pearson Correlation	0.23	0.18	-0.05	0.37
	Sig. (2-tailed)	0.16	0.27	0.78	0.02
Recursive or iterative structures (2)	Pearson Correlation	-0.29	-0.11	-0.18	-0.02
	Sig. (2-tailed)	0.07	0.48	0.27	0.89
Cascading changes (2)	Pearson Correlation	0.23	0.18	-0.05	0.37
	Sig. (2-tailed)	0.16	0.27	0.78	0.02
Mathematical modeling (2)	Pearson Correlation	0.09	0.20	0.09	0.12
	Sig. (2-tailed)	0.57	0.21	0.60	0.45
State initialization (1)	Pearson Correlation	0.24	0.42	-0.03	-0.04
	Sig. (2-tailed)	0.13	0.01	0.88	0.82
No programming errors (1)	Pearson Correlation	0.32	0.36	0.22	0.31
	Sig. (2-tailed)	0.04	0.02	0.18	0.05
Parallel Events (2)	Pearson Correlation	0.17	-0.02	0.12	-0.06
	Sig. (2-tailed)	0.30	0.90	0.45	0.72
Playthroughs yield different experiences (1)	Pearson Correlation	-0.17	0.14	0.12	0.18
	Sig. (2-tailed)	0.30	0.40	0.45	0.27
Multiple ways through game (2)	Pearson Correlation	-0.01	0.10	0.11	-0.09
	Sig. (2-tailed)	0.94	0.54	0.49	0.60
Customizable PC (1)	Pearson Correlation	-0.08	-0.09	0.06	0.09
	Sig. (2-tailed)	0.61	0.60	0.71	0.59
NPCs <> NPCs (1)	Pearson Correlation	-0.02	-0.18	-0.56	0.07
	Sig. (2-tailed)	0.90	0.27	0.00	0.69
PC <> NPC (3)	Pearson Correlation	0.01	0.22	0.23	0.09
	Sig. (2-tailed)	0.94	0.17	0.15	0.60
Information is revealed (1)	Pearson Correlation	0.17	0.08	-0.10	-0.15
	Sig. (2-tailed)	0.30	0.61	0.52	0.35
Game reference (2)	Pearson Correlation	-0.07	0.25	-0.10	0.10
	Sig. (2-tailed)	0.66	0.12	0.52	0.54
Characters animation (2)	Pearson Correlation	-0.03	0.21	0.15	0.22
	Sig. (2-tailed)	0.84	0.20	0.35	0.18
Music sound & animation (3)	Pearson Correlation	0.23	0.18	-0.05	-0.07
	Sig. (2-tailed)	0.16	0.27	0.78	0.68
Player is motivated do PC tasks(3)	Pearson Correlation	0.24	0.31	0.20	0.28
	Sig. (2-tailed)	0.13	0.05	0.23	0.08
Player takes on role (2)	Pearson Correlation	0.26	0.19	0.14	0.21
	Sig. (2-tailed)	0.10	0.24	0.37	0.20
Achievements (2)	Pearson Correlation	0.04	0.13	0.05	0.08
	Sig. (2-tailed)	0.82	0.44	0.74	0.64
Explicit rules, goals, and objectives (1)	Pearson Correlation		0.25	0.24	0.10
	Sig. (2-tailed)		0.12	0.13	0.54
Meaningful feedback (1)	Pearson Correlation	0.25		0.42	0.26
	Sig. (2-tailed)	0.12		0.01	0.10
Traditional control conventions (1)	Pearson Correlation	0.24	0.42		-0.04
	Sig. (2-tailed)	0.13	0.01		0.82
Affordance and Visibility (1)	Pearson Correlation	0.10	0.26	-0.04	
	Sig. (2-tailed)	0.54	0.10	0.82	

Summary and Implications of Results

The purpose of this research was to quantify and assess the components of CT and of GD. The *QPGDS* attempts to fulfil the deficiency in the literature by doing this. This paper outlines a preliminary use of this tool. The next section summarizes what was found in pre-service teacher's games.

Trends across Preliminary QPGDS Use

Problem Solving Opportunities (0-5). Most of the examined games contained problem solving opportunities for the player (*see Figure 2.1*). None of the games scored a perfect score of 5, and only one game scored zero points. The project that scored 0 in this category did not include any goals and just listed a set of actions for the player to complete. This category had a mean of 3.15, standard deviation of 1.00, and a median of 3.

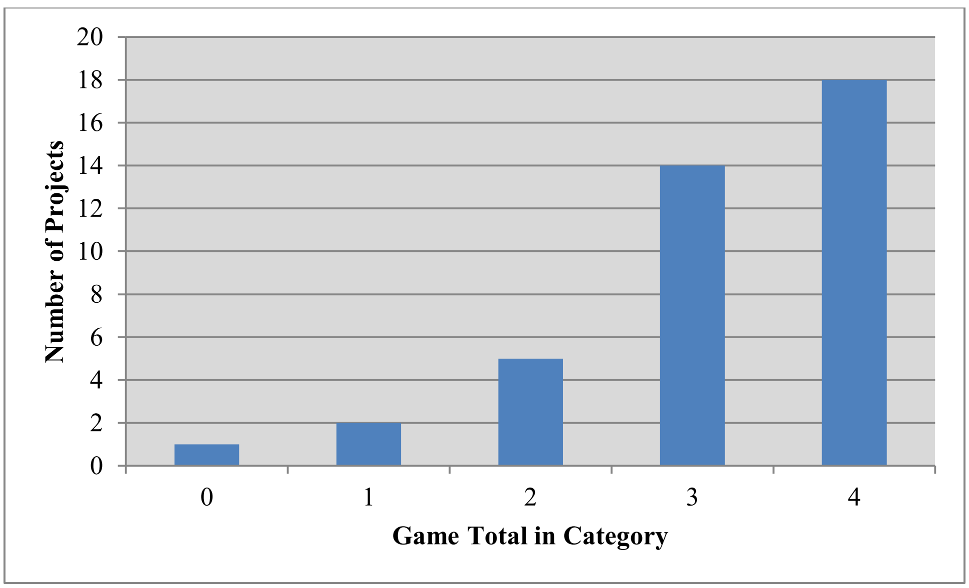


Figure 2.1. Frequency graph showing aggregated scores for games in the Problem Solving Opportunities category.

Game contains puzzles (0-2). Of all the games created, 2 games (5%) contained no puzzles, 85% offered simple puzzles (i.e., remembering, understanding, applying), and 10% contained complex puzzles requiring effort to solve (i.e., analyzing, evaluating, creating). Most of the puzzles were classified as lower-order thinking problems for the player. One of the highest scoring games contained a Mastermind-type game where the player needed to guess a sequence that the game created randomly every time based on the feedback it gave: the colour the player provided is correct and in the right spot, the colour the player provided is correct and in the wrong spot, or the colour the player provided is incorrect.

There was a strong Pearson correlation between games containing puzzles and games being forgiving ($r = 0.40$). This could be because the more challenging the game proved, the more forgiving it needed to be.

There was a strong Pearson correlation between games containing puzzles and games including goals ($r = 0.47$). This is somewhat logical, as the player needs to complete puzzles, they are simultaneously accomplishing goals.

There was a strong Pearson correlation between games containing puzzles and games providing meaningful feedback ($r = 0.61$). Games containing puzzles were more likely to give the player meaningful feedback.

There was a strong Pearson correlation between games containing puzzles and games having elements affording their function and being noticeably visible ($r = 0.47$). If games contained puzzles, they were more likely to include elements that the player knew they needed to interact with.

Forgiveness (0-1). Examining the games created, 25% of the games had a high cost of failure and were unforgiving, while most of them (75%) were forgiving. These forgiving games contained checkpoints, lives, or some system that allowed the player to try again without restarting the game. An example of an unforgiving game in an avoidance-type scenario would be if the PC touched an NPC and the player received the ‘Game Over’ message.

There was a strong Pearson correlation between games being forgiving and games providing meaningful feedback ($r = 0.66$). More forgiving games are more likely to provide the player with meaningful feedback. In a strategy-type dungeon fighter game, both the PCs and the NPCs had hit points (they took more than one hit to kill) and the characters made a sound as well as had a text bubble upon being hit.

Goals (0-1). By definition games require goals and 95% of the examined games did. Of the evaluated games, only 5% did not contain goals or objectives. One of the games that did not contain goals was a fish tank simulator. None of the player actions affects the outcome of the project.

There was a strong Pearson correlation between games including goals and games providing meaningful feedback ($r = 0.61$). If a game had goals, it was much more likely to provide the player with meaningful feedback

There was a very strong Pearson correlation between games including goals and games utilizing traditional control conventions ($r = 0.70$). Games that had goals were much more likely to use a traditional control convention.

Challenge incrementally increases as the game progresses (0-1). COT games usually progress in difficulty as the game progresses. Of the games evaluated, only 40% got more difficult as the player moves through the game; the last 60% did not increase in difficulty as the player journeyed through the game. Most of the games contained the exact same challenge throughout the game. For example, in one arcade-style shooter game, the player must shoot objects as they come towards the player's spacecraft at the bottom of the screen. The objects coming towards the player do not increase in frequency, in health, or in any difficulty at all.

Computational Thinking (0-16). None of the games scored a perfect 16 in Computational Thinking. While these conventions are taught in a CS undergraduate program, they are not always abided by in industry. Two games scored 4 points, which was the lowest total for any of the games. These games were prone to error and had poor computing style.

Figure 2.2 outlines the distribution of the game scores in Computational Thinking. This category had a mean of 8.98, standard deviation of 2.12, and a median of 9.

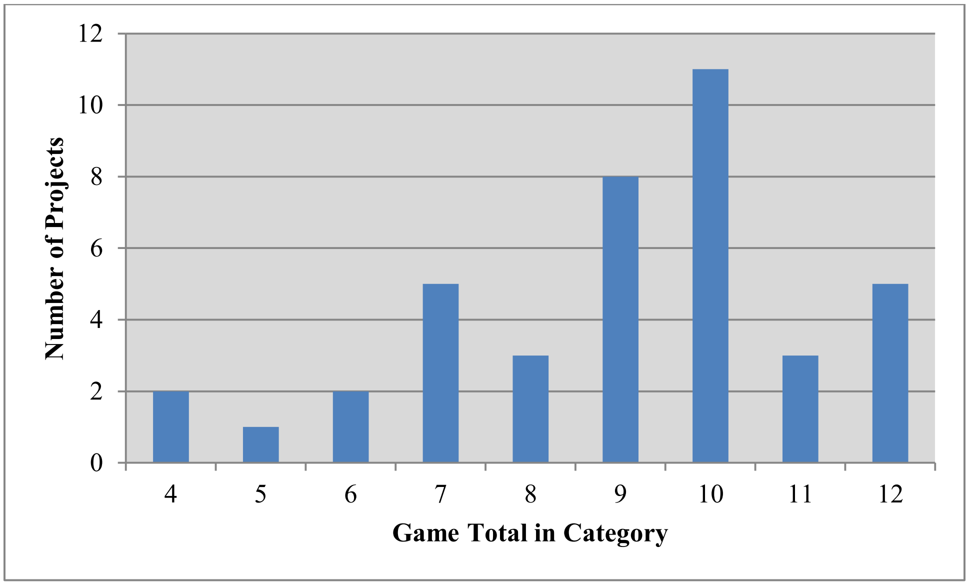


Figure 2.2. Frequency graph showing aggregated scores for games in the Computational Thinking category.

While not examined in the survey, students frequently used the default object names (e.g., Sprite 1, Sprite 2, etc.) and did not use explicit variables names (e.g., Enemy 1 life). In practice this is poor design as others looking at a programmers code can have difficulty discerning what the purpose or intention was. The original programmer may also not be aware of what they were intending to do if they look back at their code at a later time.

Another trend not examined in the survey was infinite loops. These are a computationally expensive issue that most games included. These loops, which ran forever in the game, often functionally did nothing and could have been terminated.

Modularization of game elements to a specific object (0-4). The common factor limiting games from scoring 4 points in this category was that variables were not encapsulated to the object they were pertaining to; only 18% of the games did this successfully. While local variables are supported in *Scratch* (MIT, 2009), they were not frequently used. Conversely, no games scored 0 points as code enacting a sprite appears on the object modularizing that component and making the minimum score 1 point. Figure 2.3 shows how games scored in Modularization.

While it is a feature supported in *Scratch*, very few games used local variables on their sprites. This is practical by design as some variables should only be accessed by a particular object. This reflected an inadequate understanding of encapsulation. One aspect of encapsulation that is inherently part of *Scratch* is the object-orientated nature of how code blocks pertaining to a specific sprite change that sprite from the respective sprite. Students very rarely commanded a sprite from a different one.

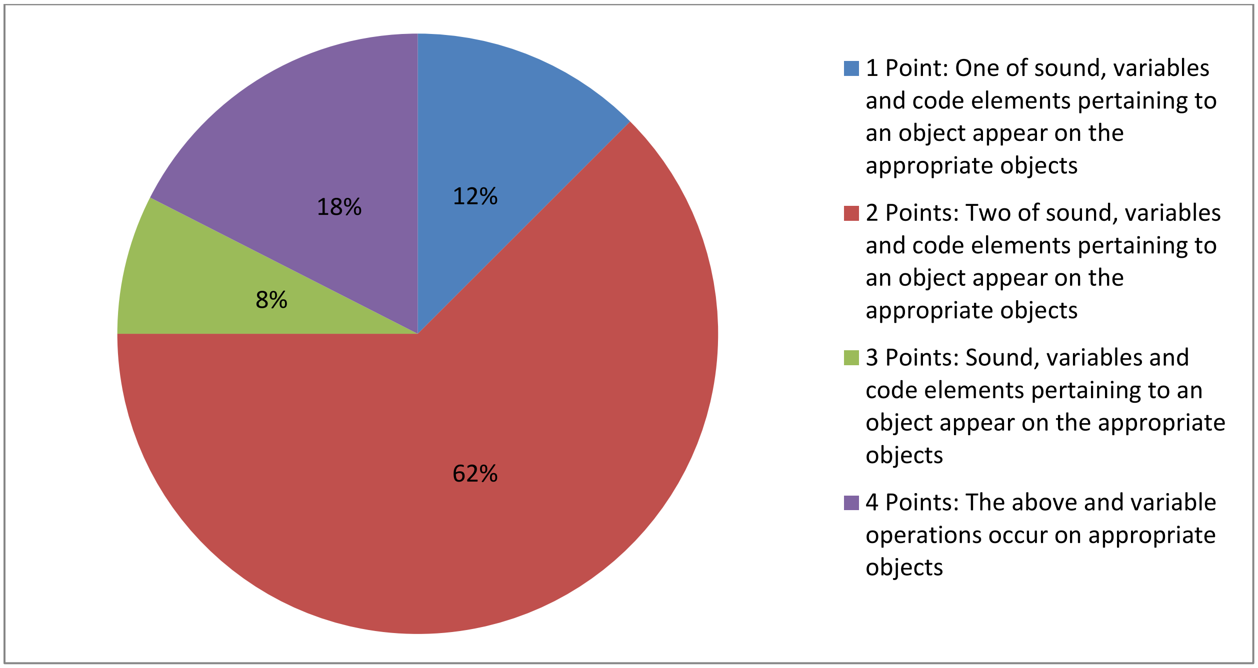


Figure 2.3. Pie chart representing percentage of games that contained modularized code.

Effective use of procedures/methods/functions (0-2). To maximize efficiency, functions should be present in a game. This allows for code reuse at later times; 70% of the projects were refactored to include functions. Games can run without having functions through using variables, but this utilizes loops checking constantly throughout the game and requiring additional computational power. Of the games examined, 7.5% of them did not have any functions. Additionally, this set can include games that were not very complex and did not require variables or functions. The pre-service teacher that created one of the games that did not use functions wrote in their reflection that they preferred to use variables over functions. Some of the games (22%) contained some functions, but had noticeable areas that could have been refactored into functions.

There was a very strong Pearson correlation between games utilizing procedures/methods and games cascading changes (r = 1.00). Students that optimally used procedures in their game were much more likely to properly cascade changes to different objects in their games.

There was a strong Pearson correlation between games utilizing procedures/methods and PC interactions with the NPC (r = 0.41). Games that utilized procedures were more likely to have PC interactions with NPCs.

Recursive/iterative vs. linear structures (0-2). Roughly half of the games (55%) contained iterative or recursive structures wherever optimal. The other half (45%) utilized linear code structures instead of recursive or iterative structures. This is less efficient as the programmer is doing work that the computer can do. It is also less organized as there is more code on the screen to read through. Quite commonly specific sprite costumes were specified linearly instead of choosing the next costume based on the costume number variable. This reflects a lesser understanding of the code's processes (*see Figure 2.4*).

Figure 2.4. A linear section of code that could have been refactored to omit redundancies. The SWITCH TO COSTUME… blocks could be replaced in a REPEAT loop with NEXT COSTUME inside.

Cascading changes driven from one location (0-2). Most of the games (70%) cascaded state changes from a central point. A minority of the games (7.5%) were redundantly making constant checks in loops to verify if they needed to change state. An example of this would be having a FOREVER loop with multiple IF conditions inside. Some of the games (22.5%) contained properties consistent with both of the above.

There was a strong Pearson correlation between games cascading changes driven from one location and PC interaction with NPCs (r = 0.41). Games driving state changes from one central location were more likely to have PC interactions with NPCs.

Accurate use of mathematical modeling (0-2). Higher order games contain accurate mathematical models. Only one of the examined games (2.5%) contained an accurate mathematical model. A key mechanism in this game was a ball bouncing around and the bounce trajectories were modelled accurately. No mathematical models were present or effectively demonstrated in 77.5% of the games, whereas models were present, but not accurately modeled in 20% of the games (e.g., having a character's jump hardcoded to include an up and down component instead of a constant force pushing the character down unless it is touching the ground).

There was a strong Pearson correlation between games including accurate mathematical modeling and games including customizable PC features (r = 0.52). If an accurate mathematical model appeared in a game, that game was more likely to have customizable PC features.

Proper state initialization (0-1). Proper state initialization is important to having the game function as the designer intends. Without proper state initialization, the player can be on the final level of the game when they start the game. Only one of the games was not properly initialized (2.5%). This game had sprites in the wrong place when the game started. Most of the students recognized that this is an important part of game design (97.5%).

There was a strong Pearson correlation between games with proper state initialization and games having music, sound, and animation ($r = 0.56$). Games with proper state initialization were more likely to have more of music, sound, and animation. Students were strongly encouraged to pay attention to game state initialization and to include music, sound, and animation in their games.

There was a strong Pearson correlation between games with proper state initialization and games providing meaningful feedback ($r = 0.42$). Games with proper state initialization were more likely to provide meaningful feedback.

Game play is clean with at most minor programming errors (0-1). Pre-service teachers usually enjoyed making their games, but they did have a finite amount of time to work on them (in addition to their other classes). Of the games examined, 65% played through smoothly without major programming errors, whereas 35% included major errors that made playing through the game difficult or impossible. An example of an error includes getting stuck in a state that made it impossible to continue on with the game.

There was a strong negative Pearson correlation between games having no errors and games running smoothly in parallel ($r = -0.41$). Games that had no errors were more likely to have processes that ran smoothly in parallel. Games that had game-breaking errors were more likely to have other errors that were not game breaking (a component of events running smoothly in parallel).

Events are driven smoothly in parallel (0-2). One challenging concept in programming is that of parallel processes. Games are driven by events and frequently there are multiple operations happening in the game that must work in parallel. While not necessarily game-breaking errors, 37.5% of the games contained noticeable errors. These errors were frequently game elements, including NPCs, remaining on the screen when the state changes. The other portion (63.5%) of the games contained processes flawlessly working in parallel with one another.

Customization of Player Experience (0-4). A majority of the games played through linearly and did not allow for much customization of the player experience (*see Figure 2.5*). A game that scored high in this category was a dungeon fighter strategy game. The player could choose which permutation, or individual, out of the warrior, wizard, priest, and witch they wished to use in combat. An example of a game that scored low in this category was a demon hunter game. The player is given one character resulting in the same experience every playthrough. This category had a mean of 2.35, standard deviation of 0.95, and a median of 2.

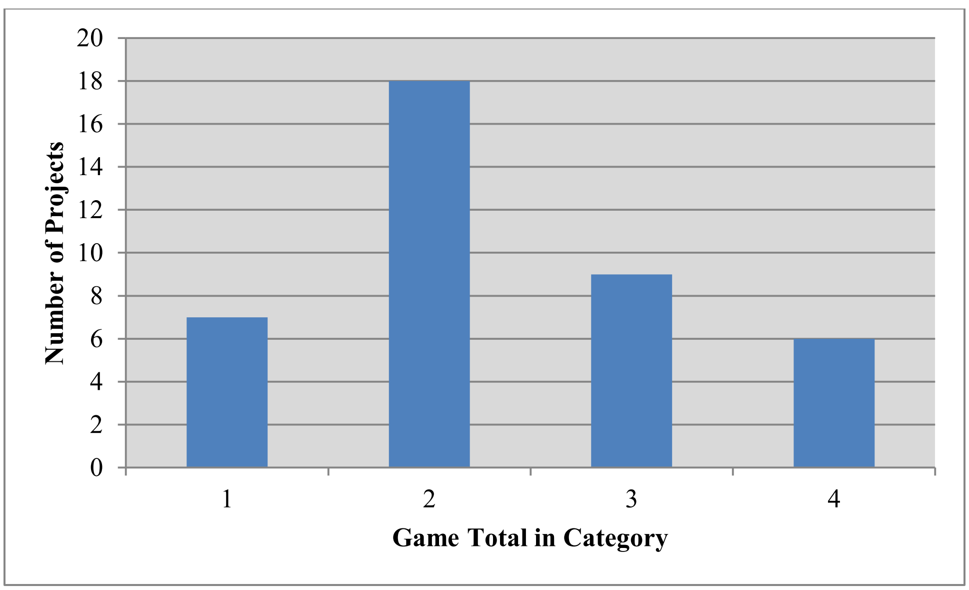


Figure 2.5. Frequency graph showing aggregated scores for games in Customization of Player Experience category.

Multiple playthroughs yield different experiences (0-1). Of the games examined, 62.5% were the same on subsequent playthroughs. Alternatively, 37.5% offered the player a different experience if played again.

This criterion was found to have a moderate correlative relationship with the next category. While similar, this category can include a dynamic experience from the game artificial intelligence through the use of random actions undertaken by the NPCs.

Allows multiple ways through the game based on player choices (0-2). Of the games examined, the player was not given opportunities to make choices throughout the game 67.5% of the time resulting in a linear experience. The player was afforded choices affecting the outcome of the game 32.5% of the time.

Players can customize the Player Character (0-1). The pre-service teachers only allowed the PC to be customizable in 12.5% of the created games. Conversely, 87.5% of the games did not allow for any player customization. An example of a game that scored the full point in this category is a Batman arcade fighter game; the player can choose whether they would like to play as Batman or Robin. Each of these two characters offers a different playing style.

Player Interaction (0-7). Most games fell under the lower-middle range of this category (*See Figure 2.6*). The game that scored no points in this category was a multiplayer game that offered no features for single player mode and did not offer any information revealing for the controls or goals of the game. This category had a mean of 2.5, standard deviation of 1.28, and a median of 2.

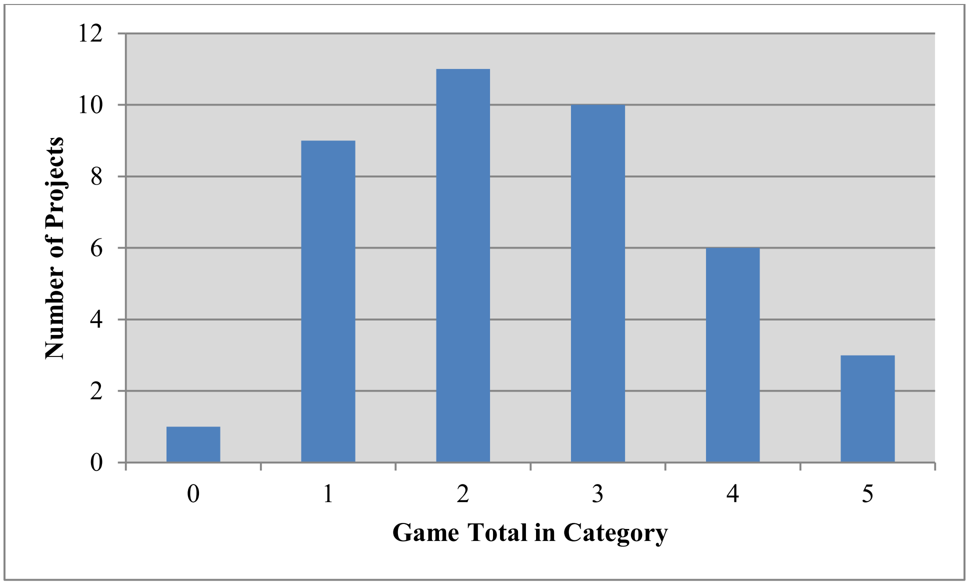


Figure 2.6. Frequency graph showing aggregated scores for games in Player Interaction category.

NPC interactions with other NPCs (0-1). Few games featured NPC interactions with other NPCs (7.5%). The only Sports-type game in the dataset had a football theme and featured multiple mini games. A couple of these mini games included friendly NPCs to help the player block the other team as the PC quarterback attempts to run the ball in for a touchdown. Quite often NPCs had no effect on each other (92.5%). In the Batman arcade fighter game, the NPCs run through one another and do not register each other's presence.

There was a strong negative Pearson correlation between games having NPC interactions with other NPCs and games using traditional control conventions (r = -0.56). If a game had NPC interactions with other NPCs, it was less likely to use traditional control conventions.

Player interactions with NPCs (0-3). For the most part, NPCs functioned linearly doing or saying the same things throughout the game (52.5%). Next, 32.5% of the games did not feature NPC. A small subset (12.5%) of the games included interactions with NPCs that changed throughout the game. Only one game (2.5%) had NPCs that reacted to the player's actions. In this game, the NPCs' actions (e.g., attack times, where they move, what they say) depended on the PC's actions and positions. Figure 2.7 outlines project scores on player interactions with NPCs.

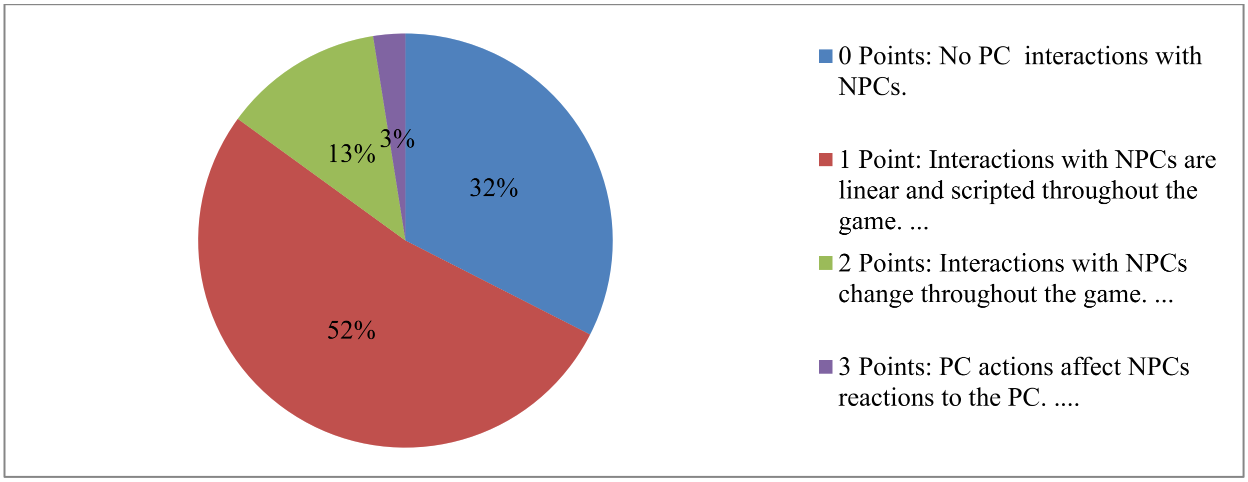


Figure 2.7. Pie chart representing percentages of games had different levels of player interactions with NPC.

Information is revealed as the player needs it (0-2). A subset of the games did not provide information as the player needed it (30%). These games made the player guess the controls and the game goals. Most of the games told the player explicitly what needed to be done (60%). These games frequently gave specific instructions without any narrative context.

There was a strong Pearson correlation between games revealing information to the player and games having a reference that the user can look at if they desire ($r = 0.52$). Games that revealed information to the player were more likely to have a reference that the user could look at if they desire. More generally, games that provide information to the player were more likely to have additional resources that the player could consult.

A reference can be looked at if the user desires (0-2). A fair number of games (30%) did not include any reference for the player; in these games the player needed to guess what the controls were. In 62.5% of the games, player controls or hints are available sometime in the game, but are not always accessible should the player need to refer to them again. In the last 7.5% of the games, the player can lookup controls, or even hints whenever they need a reminder at any point in the game. The football Sports-themed mini game collection had a help button on every screen to let the player get more information about the goals that was selectable at any time.

There was a strong Pearson correlation between games including a reference the user can look at if they desire and games including character animations (r = 0.51). Games including an always available reference were more likely to include character animations.

Player Immersion (0-5). Most games fell in the lower middle range of this category scores (*See Figure 2.8*). This category could also directly be interpreted as how much time the designer spent on animation, sound, and music. The higher the score, the more of these features the game includes. This category had a mean of 2.65, standard deviation of 1.53, and a median of 2.5.

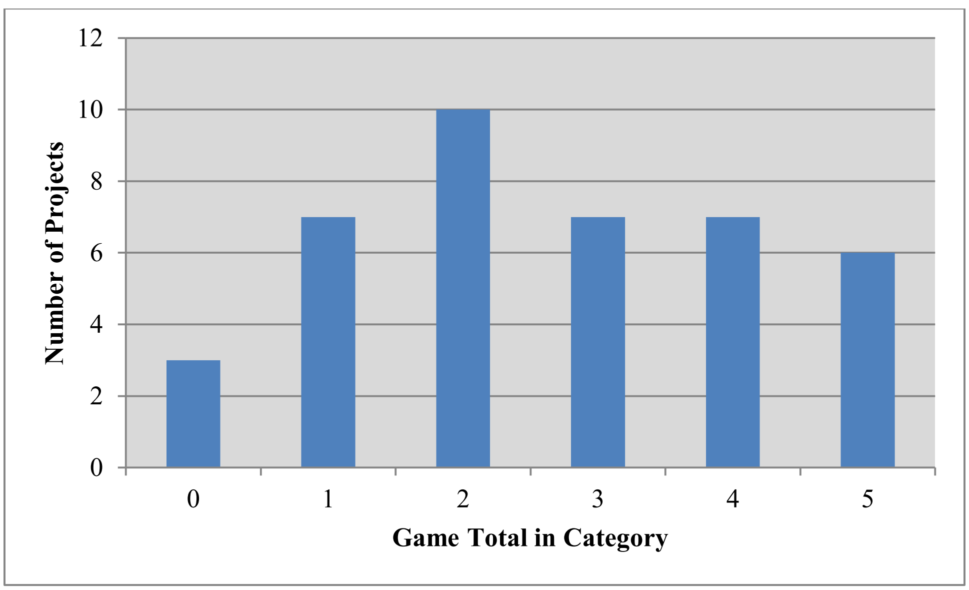


Figure 2.8. Frequency graph showing aggregated scores for games in Player Immersion category.

Characters animation (0-2). In roughly half the games (52.5%) of the games, the characters are not animated at all. In less of the games (30%), only the PC is animated. In a smaller subset (17.5%) of the games, the NPCs and PCs are animated. Figure 2.9 outlines this relationship. The Batman fighter arcade game contained animations for all the characters that appeared in the game.

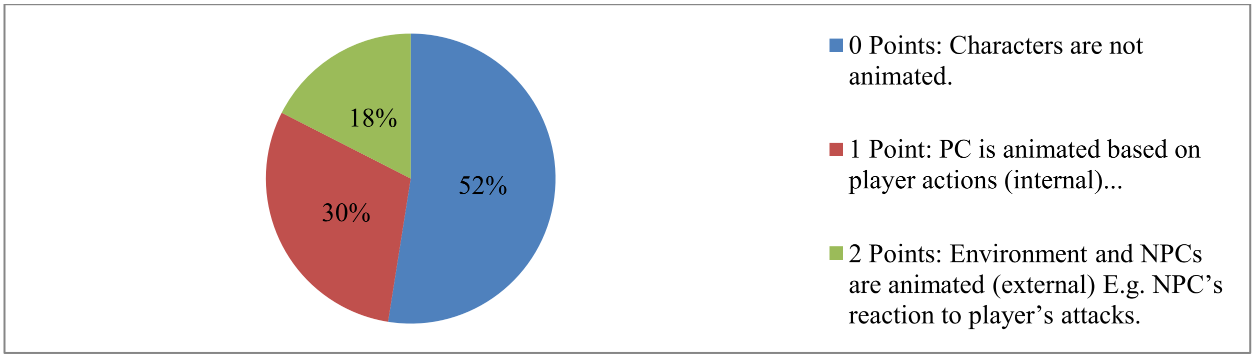


Figure 2.9. Pie chart representing percentages of games had different levels of character animation.

Music, sound, and animation (0-3). A majority of the games examined contained at least two components of music, sound, and animation creating one central thematic experience for the player. Only two games (7.5%) surveyed did not have any of these three. Figure 2.10 outlines the percentage of games to contain music, sound, and animation.

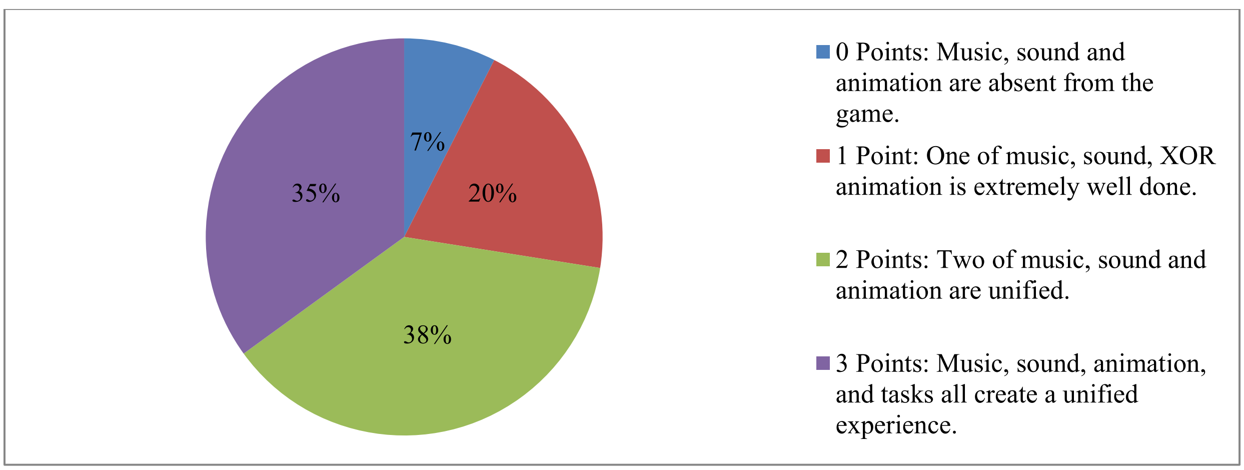


Figure 2.10. Pie chart representing percentages of games had different levels of music, sound, and animation used in the game.

Player Motivation (0-8). Quite often the player did not actually take control of a particular character. No narrative was developed and the player was thrust into a situation where goals were unclear. In these cases, rules were implicit and the player had to guess what needed to be done without a supporting narrative or instructions. A little less commonly, the player would take control of a persona that was not developed throughout the game. In more engaging games, the PC grows as the game progresses. The dungeon strategy game was the only game to score full points in this category. Figure 2.11 outlines the frequency of game scores for the Player Motivation category. This category had a mean of 2.38, standard deviation of 1.76, and a median of 2.

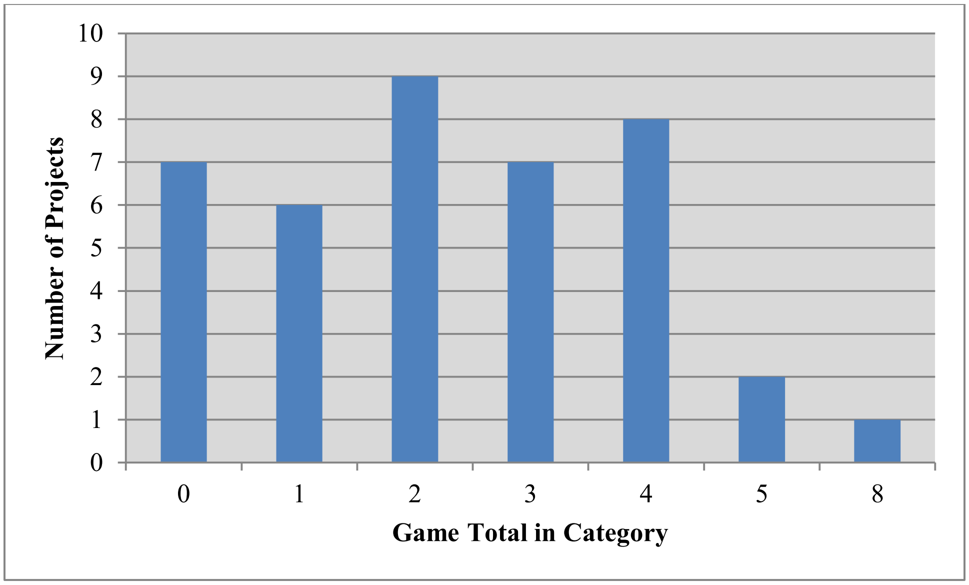


Figure 2.11. Frequency graph showing aggregated scores for games in Player Motivation category.

Player is motivated to complete tasks given to PC (0-3). Only a couple games contained storylines that were developed throughout the game (7%; *See Figure 2.12*). Of the games that had stories, most were only present at the beginning of the game to set up a theme and were not developed (25%).

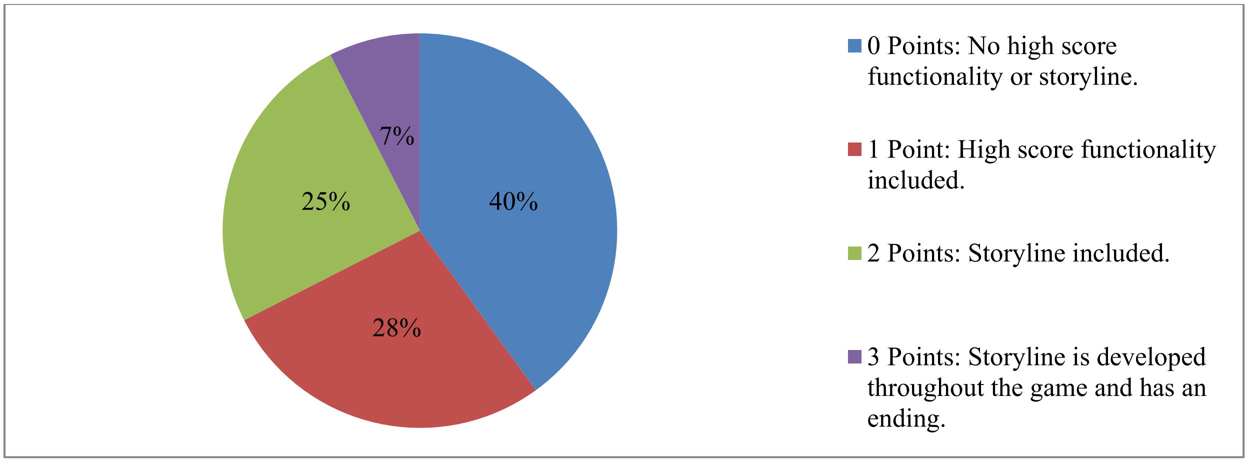


Figure 2.12. Pie chart representing percentages of games motivating the player by high score functionality or narrative development.

Users assume a role in the game, rather than simply playing (0-2). Over half of the games (55%) did not require the player to take on a role. Instead the player is abruptly given control over a character without a back-story. Additionally, 40% of the time, players were able to create or take on a persona that was not developed throughout the game. Conversely, PC personas were only developed in two of the games (5%).

Achievements present in the game (0-2). Achievements are a major component of COT games. They provide additional depth and challenge for players. Only three (7.5%) of the examined games contained achievements, whereas one (2.5%) contained achievements in the way that the player was not alerted when they accomplished the feat. A Country music trivia game contained achievements that appeared when the player scored over a certain threshold score. A majority of the games (90%) did not contain any achievements at all.

Rules, goals and objectives are explicit (0-1). Most of the games contained goals that were explicit (70%), but 30% of the games did not make their goals explicit. Games with non-explicit goals dropped the player into a scenario where they have to guess what to do.

Interface Usability (0-3). Most pre-service teachers inherently created games and interfaces that were intuitive to use and followed good interface guidelines (*See Figure 2.13*).

The one game that did not include an intuitive interface included many sprites and the goals were not obvious. This category had a mean of 2.8, standard deviation of 0.52, and a median of 3.

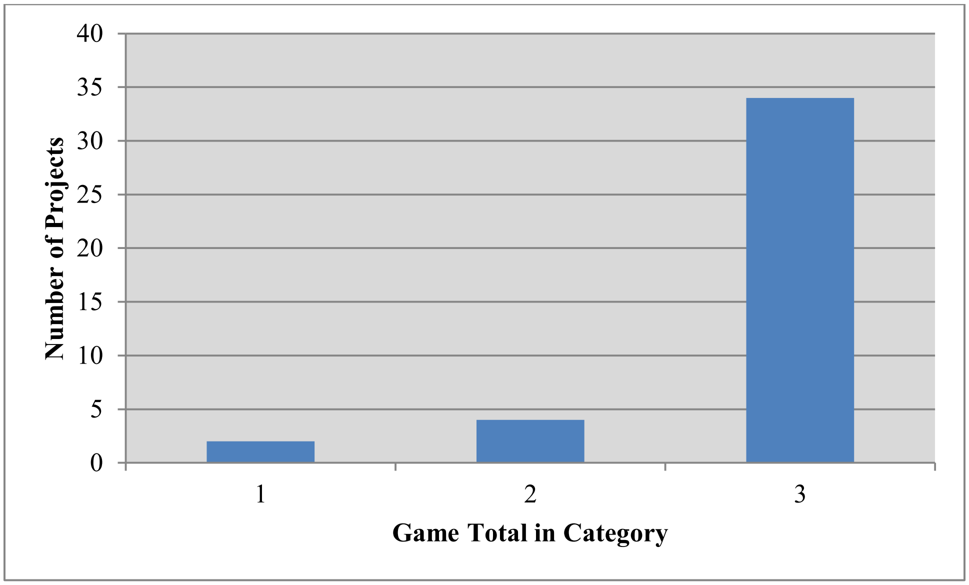


Figure 2.13. Frequency graph showing aggregated scores for games in Interface Usability category.

Meaningful feedback (0-1). Most games provided feedback to the player when there was a change of state (87.5%). Frequently this was done by having an animation when the player died or having a sound play as a projectile was shot. Only five of the forty (12.5%) games did not provide meaningful feedback when there was a change of state.

There was a strong Pearson correlation between games that presented meaningful feedback and games including traditional control conventions ($r = 0.42$). Games that provided the player with meaningful feedback were more likely to use traditional control conventions

Uses traditional control conventions (0-1). Only one game (2.5%) did not use an existing control scheme when it made sense to use one. This game used a strange combination of keyboard keys and mouse clicks that did not enhance the game in any manner. Almost all of the games (97.5%) used traditional game control conventions. The arrow keys, ASDW, or the mouse was used in these cases.

Affordance and visibility (0-1). The objects that the player was supposed to interact with or avoid were almost always apparent (95%) of the time. Only in two instances was this not discernible (5%). In these instances the screen was cluttered with sprites and the ones the player is supposed to manipulate were not clear.

Discussion

With the advent of new user-friendly tools for game creation, teachers can now empower their K-12 students with the possibility of learning through designing and building their own personal digital games. The game construction process can offer these students alternative ways to represent and instantiate their knowledge of a content area while supporting their learning of computational thinking skills. For teacher education programs to help prepare pre-service teachers to effectively use game construction in classroom practice it important to provide them with a deeper understanding of how games are designed and built. A situated, hands-on constructionist learning experience is one way this can be accomplished. In this situated context the pre-service teacher is an apprentice designer and digital game builder, thereby learning important skills.

An important question that guided this study was: How can the CT and GD skills important for pre-service teachers be quantitatively assessed? Identifying these skills can prepare pre-service teacher programs so they can prepare the professionals of tomorrow.

This paper examined a purposeful sample of the digital games created by the pre-service teachers enrolled in an elective Educational Technology course. They were free to build a game of their choice. An important goal of the course was learn digital game construction skills that could be used to support and enhance their future K-12 teaching. The main aim of this research was to quantify CT and GD practices. We have proposed the *Quality Practices of Game Design*

Survey to meet the deficiency in the literature (*see Table 2.4*). What follows is a discussion of results found in the seven QPGDS categories.

The Problem Solving Opportunities category, whose purpose is to engage the player with challenges, was scored out of 5 points ($\mu = 3.15$, $\sigma = 1.00$, $X_{(1)} = 0$, $X_{(n)} = 4$). Most games (95%) contained solving opportunities for the player, but puzzles were frequently simple (e.g., remembering, understanding, applying). Games were designed to be forgiving; 75% of them had a system in place so the player did not need to restart from the beginning (e.g., checkpoints, lives). Almost all of the games contained goals (95%) and a very strong correlative relationship was identified between games including goals and games utilizing traditional control conventions ($r = 0.70$). Games that had goals were much more likely to use a traditional control convention. Sixty percent of the games did not increase in difficulty as the player progressed through the game.

The Computational Thinking category, which measures the designer's CT proficiency, was scored out of 16 points ($\mu = 8.98$, $\sigma = 2.12$, $X_{(1)} = 4$, $X_{(n)} = 12$). The lowest scoring games were prone to error, and had poor programming style; whereas the opposite was true for the highest scoring games; the highest scoring games' code was also easier to interpret. While examining encapsulation of code, most projects only modularized two of sound, variables, and code on the appropriate objects (62%). Maximum programming style requires code that appears frequently to appear as functions to be reused in games and 70% of the games did this. There was a very strong correlation between games utilizing procedures/methods and games cascading changes ($r = 1.00$); students that used optimally used procedures in their game were much more likely to properly cascade changes to different objects in their games. Roughly half the games contained iterative or recursive structures wherever optimal, while the other half were not fully optimized, thereby reflecting a lesser understanding of the code's process. Most games (70%)

cascaded change from one location to another. Mathematical models were present in 80% of the games examined. Most students recognized the importance of initializing state properly (97.5%). Most of the games (65%) played smoothly without major programming errors, whereas 35% included major errors making game play difficult or impossible. There was a strong negative Pearson correlation between games having no errors and games running smoothly in parallel (r = -0.41). Games that had no errors were more likely to have processes that ran smoothly in parallel. Games that had game-breaking errors were more likely to have other errors that were not game breaking (a component of events running smoothly in parallel). While not necessarily game-breaking errors, 37.5% of the games contained noticeable errors. These errors were frequently game elements, including NPCs, remaining on the screen when the state changes. The other portion (63.5%) of the games contained processes flawlessly working in parallel with one another.

Two items not explored by the survey relating to Computational Thinking are giving explicit names to variables and objects, and infinite loops. Variables and objects frequently did not have explicit names (e.g., Sprite 1, Sprite 2, life). This makes programs more difficult for programmers to interpret as some context is required, while the infinite loops are computationally expensive and inefficient.

The Customization of Player Experience category, which allows player to have a personalized experience, was scored out of 4 points ($\mu = 2.35$, $\sigma = 0.95$, $X_{(1)} = 1$, $X_{(n)} = 4$). Most of the games did not allow for much customization of the player experience. Of the games examined, 62.5% were the same on subsequent playthroughs. Most games (67.5%) featured one way to get through the game, resulting in a linear experience upon replay. The PC was not customizable in 87.5% of the games examined. Customization of the PC had a moderate correlation with there being multiple ways through the game ($r = 0.38$). The pre-service teachers that had multiple ways through their game were more likely to include a customizable PC.

The Player Interaction category, which measured the interactivity of the game, was scored out of 7 points ($\mu = 2.5$, $\sigma = 1.28$, $X_{(1)} = 0$, $X_{(n)} = 5$). Quite often NPCs had no effect on each other (92.5%). If games featured NPCs (32.5% did not), 52.5% of them functioned linearly throughout the game by saying or doing the same thing. In 60% of the games the players were told explicitly what needed to be done instead of shown, or not told at all. In 62.5% of the games, player controls were available sometime, but not at all times; in 30% of the games no reference was provided for the player, whereas in 7.5% of the games player could access hints or controls at all times. There was a strong correlative relationship between a game reference being available and information being revealed to the player ($r = 0.52$). The more accessible control and hint information was to the player, the more likely puzzle solutions were to be communicated implicitly.

The Player Immersion category, which contributed to a polished product, was scored out of 5 points ($\mu = 2.65$, $\sigma = 1.53$, $X_{(1)} =0$, $X_{(n)} = 5$). The higher the score in this category, the more attention the designer applied to animation, sound, and music. Game characters were not animated in 52.5% of the games, whereas only the PC was animated in 30% of the games. A majority of the games (65%) examined contained at least two of music, sound, and animation creating one central thematic experience for the player. There was a strong correlative relationship between a game reference being available and the level of character animations in the game ($r = 0.51$). If a game reference was available, the characters in the game were more likely to be animated.

The Player Motivation category, which measures player interest in completing tasks, was scored out of 8 points ($\mu = 2.38$, $\sigma = 1.76$, $X_{(1)} = 0$, $X_{(n)} = 8$). Frequently there was no high score or storyline (40%). In 28% of the games examined, there is only high score functionality, while 25% only have a storyline that is not developed throughout the game. Only 7% of the games have a plot that is developed throughout the game. Quite often the player simply plays without a role (55%). A majority of the games do not contain achievements (90%). Rules, goals, objectives were explicit in 70% of the games.

The Interface Usability category was scored out of 3 points ($\mu = 2.8$, $\sigma = 0.52$, $X_{(1)} = 1$, $X_{(n)} = 3$). Most pre-service teachers inherently created games and interfaces that were intuitive to use and followed good interface guidelines. Most of the games (87.5%) provided meaningful feedback when there was a change of state. Almost all of them (97.5%) implemented a traditional control scheme for the player. Objects that the player is supposed to interact with were apparent in 95% of the games. Games providing meaningful feedback to the player were more likely to have a low cost of failure as there was a strong correlative relationship between the two items ($r = 0.66$).

Gaming, both digital and not, provides players and creators an engaging opportunity to develop conditional logic, algorithm building, debugging, simulation, and distributed computation skills (Berland & Lee, 2011). Brennan and Resnick (2014) write that quality assessment allows for the incorporation of artifacts and values multiple ways of knowing. Constructionism empowers learners allowing them to have agency in their learning experience (Paper, 1971).

GD, a constructionist experience, could be priceless in education. COT games incorporate some of education's best conventions. Learning good GD could benefit pre-service teachers as they are learning conventions they could integrate in their classrooms. One of the most important principles is forgiveness (or low cost of failure; Gee, 2005; Gee, 2013). If students understand that the cost of failure is lowered, they may be more willing to be actively engaged in class or test out new patterns of thinking. Once producing a new pattern of thinking, it is important that students are provided with meaningful, immediate feedback, and affirmation of performance (Dickey, 2005). This follows reinforcement theory; if a students' understanding is correct, educators can reinforce this response by immediately confirming a students' understanding. Conversely, if the response is unsuitable, educators can provide meaningful feedback to help students achieve a correct comprehension that they can exhibit in subsequent scenarios.

Giving a student or player control over what they are doing makes them more involved in the learning process awards them more control over their experience (Dickey, 2005; Gee, 2005; Gee; 2013; Mayer, 2011). These quality principles of GD were summarized in Table 2.3.

A developed CT skill set is beneficial to any profession in education. Algorithmic thinking (Ater-Kranov, Bryant, Orr, Wallace, & Zhang, 2010) allows for any process to be analyzed and optimized. Different processes that appear in any given algorithm can be modularized or abstracted in order to simplify procedures further (Wing, 2006). Computational concepts requiring logic are used on a daily basis, and a formal understanding of these constructs can assist with questioning daily procedures (Brennan & Resnick, 2012). Table 2.2 summarizes the components of CT identified in literature.

A growing number of countries in the world are reforming their education systems to include CT and CS (Furber, 2013; PITAC, 2005; State Government of California, 2014). In order to cultivate a generation with CT skills, teachers need to be properly prepared before they can familiarize students with CT throughout their K-12 education (Barr and Stephenson, 2011).

In the near future, CT will impact everyone, in every field, and will be a fundamental skill used by everyone (Wing, 2006). While CT is applicable to all disciplines, it is primarily and exclusively taught through CS in today's schools, if taught at all. Harel and Papert (1991) maintain that programming is reflexive with other domains: learning to program can be easier to learn in conjunction with another domain. Learning in the Information Age must include a fusion of technological fluency and content preparing students to work and live in the Information Age (Denton, 2012)

Our study has a few limitations that should be addressed in future research. First, our sample is not representative of all pre-service teachers. While our sample consisted of both pre-service teachers in an elementary and secondary traditional four-year undergraduate teacher programs, this sample was based on those students who selected this educational technology course as an education course elective. Therefore, they choose this class based on their interest in the area

Interestingly, with respect to the number of pre-service teacher males and females in our study, we had more males than females. The overall demographic students in the pre-service teacher program at our post-secondary institution are conversely 77% female and 23% male. It can be concluded that perhaps males are more interested in creating digital game multimedia over their female counterparts. Possible next steps would be applying the QPDGDS tool to a larger dataset, refining the items in the QPGDS categories to reflect a higher degree of sensitivity, and applying the QPGDS to assess games created in other programming environments.

References

Adams, E. (2007). *Fundamentals of game design*. Upper Saddle River N.J.: Pearson Prentice Hall.

Alberta Education. (2014). *Programs of study*. Retrieved from http://education.alberta.ca/teachers/program.aspx

AllGame. (2010). *Game Genres.* Retrieved from http://www.allgame.com/genres.php

Anton, G., Harris, S., Ochsner, A., Rothschild, M., & Squire, K. (2013). Patterns of play: Understanding computational thinking through game design. In *Proceedings of the Games, Learning, and Society Conference* (Vol. 3). Retrieved from http://press.etc.cmu.edu/files/GLS9.0-Proceedings-2013-web.pdf

Ater-Kranov, A., Bryant, R., Orr, G., Wallace, S., & Zhang, M. (2010). Developing a community definition and teaching modules for computational thinking: accomplishments and challenges. In *Proceedings of the 2010 ACM conference on Information technology education* (pp. 143-148). ACM. doi: 10.1145/1867651

Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *Learning & Leading with Technology*, *38*(6), 20-23. Retrieved from http://www.learningandleading-digital.com/learningandleading

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54. doi: 10.1145/1929887.1929905

Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010, June). Using scalable game design to teach computer science from middle school to graduate school. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 224-228). ACM. doi: 10.1145/1822090.1822154

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011, March). Recognizing computational thinking patterns. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 245-250). ACM. doi: 10.1145/1953163.1953241

Berland, M., & Lee, V. R. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, *1*(2), 65. doi: 10.4018/ijgbl.2011040105

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145-157. doi: 10.1016/j.compedu.2013.10.020

Bloom, B. S. (1956). Taxonomy of educational objectives. Vol. 1: Cognitive domain. *New York: McKay*.

Blum, L., & Cortina, T. J. (2007, March). CS4HS: an outreach program for high school CS teachers. In *ACM SIGCSE Bulletin* (Vol. 39, No. 1, pp. 19-23). ACM. doi: 10.1145/1227504.1227320

Boechler, P., Artym, C., DeJong, E., Carbonaro, M., & Stroulia, E. (2014). Computational thinking, code complexity and prior experience in a videogame-building assignment. *Proceedings of the IEEE International Conference on Advanced Learning Technologies,* Athens, Greece. doi: 10.1109/ICALT.2014.118

Boechler, P., Carbonaro, M., deJong, E., Stroulia, E. & Gutiérrez, L. (2011). Video games for teaching and learning: game-building as a teaching tool. In M. Koehler & P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2011* (pp. 2054-2061). Chesapeake, VA: AACE. Retrieved June 25, 2014 from http://www.editlib.org/p/36606

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association,* Vancouver, Canada. Retrieved from http://www.aera.net/

Bruckman, A., and Resnick, M. (1995). *The MediaMOO Project: Constructionism and Professional Community. Convergence*, 1(1): 94-109. doi: 10.1177/135485659500100110

California Legislative Information. (2014). *Bill search*. Retrieved from https://leginfo.legislature.ca.gov/faces/billSearchClient.xhtml

Carbonaro, M. (1997). Making technology an integral part of teaching: The development of a constructionist multimedia course for teacher education. *Journal of Technology and Teacher Education*, *5*(4), 255-280. Retrieved from http://dl.acm.org/citation.cfm?id=274219

Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: A gender-neutral attractor to computing science. *Computers & Education*, 55(3), 1098-1111. doi: 10.1016/j.compedu.2010.05.007

Chambers, J. & Carbonaro, M. (2003). Designing, developing, and implementing a course on lego robotics for technology teacher education. *Journal of Technology and Teacher Education*, 11(2), 209-241. Norfolk, VA: AACE. Retrieved from http://site.aace.org/pubs/jtate/default.html

Code. (2014). *Leaders and trend-setters all agree on one thing*. Retrieved from http://code.org/quotes

Columbia Journalism School. (February 10, 2014). *Computational thinking with jeannette wing* [Video file]. Retrieved from https://www.youtube.com/watch?v=U67utvZai8s

Denning, P. J. (2009). The profession of IT beyond computational thinking. *Communications of the ACM*, *52*(6), 28-30. doi: 10.1145/1516046.1516054

Denton, D. W. (2012). Enhancing Instruction through Constructivism, Cooperative Learning, and Cloud Computing. *TechTrends*, 56(4), 34-41. doi: 10.1007/s11528-012-0585-1

Dickey, M. D. (2005). Engaging by design: How engagement strategies in popular computer and video games can inform instructional design. *Educational Technology Research and Development*, 53(2), 67–83. doi: 10.1007/bf02504866

Elliott, L., Ream, G., McGinsky, E., & Dunlap, E. (2012). The contribution of game genre and other use patterns to problem video game play among adult video gamers. *International journal of mental health and addiction*, 10(6), 948-969. doi: 10.1007/s11469-012-9391-4

ESA, the. (2013). *Demographic and usage data essential facts about the computer and video game industry*. Retrieved from http://theesa.ca/wp-content/uploads/2013/10/Essential-Facts-English.pdf

Extra Credits. (January 22, 2014a). *Extra credits - the magic circle - how games transport us to new worlds*. [Video file]. Retrieved from http://youtu.be/qZ-EY9gTsgU

Extra Credits. (January 29, 2014b). *Extra credits - affordances - how design teaches us without words*. [Video file]. Retrieved from http://youtu.be/QCSXEKHL6fc

Federici, S. & Stern, L. (2011). A constructionist approach to computer science. In S. Barton et al. (Eds.), *Proceedings of Global Learn 2011* (pp. 1352-1361). AACE. Retrieved from http://www.aace.org/

Ford, J. L. (2008). *Scratch Programming for Teens*. Course Technology PTR, Boston MA.

Furber, S. (2012). Shut down or restart? The way forward for computing in UK schools. *The Royal Society, London*. Retrieved from https://royalsociety.org/education/policy/computing-in-schools/report/

GameFAQs. (2014). *Category Search*. Retrieved from http://www.gamefaqs.com/search/index.html?game=genre

GameFly, Inc. (2003, February 28). *Genre Definitions*. Retrieved from http://www.mobygames.com/glossary/genres/

Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York, NY: Palgrave Macmillan. doi: 10.1145/950566.950595

Gee, J. P. (2005). Good video games and good learning. In *Phi Kappa Phi Forum* (Vol. 85, No. 2, p. 33). THE HONOR SOCIETY OF PHI KAPPA PHI. Retrieved from http://www.phikappaphi.org/publications-resources/

Gee, J. P. (2006). Why game studies now? Video games: A new art form. *Games and culture*, *1*(1), 58-61. Retrieved from http://www.jamespaulgee.com/sites/default/files/pub/GamesCulture-1-1.pdf

Gee, J. P. (2007). *Good video games+ good learning: Collected essays on video games, learning, and literacy*. New York: P. Lang. doi: 10.1177/1555412005281788

Gee, J.P. (2013). Video games and learning [Video file].

Good, J. (2011). Learners at the wheel: Novice programming environments come of age. *International Journal of People-Oriented Programming (IJPOP)*, *1*(1), 1-24. doi: 10.4018/ijpop.2011010101

Google. (2010). *Exploring Computational Thinking*. Retrieved from http://www.google.com/edu/computational-thinking/

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. doi: 10.3102/0013189x12463051

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25-27. doi: 10.1145/1378704.1378713

Hamlen, K. R.(2011). Children’s choices and strategies in video games. *Computers in Human Behavior*. January 2011, 27 (1), p. 532-539. doi: 10.1016/j.chb.2010.10.001

Harel, I. (1988). *Software design for learning: children's construction of meaning for fractions and LOGO programming*. Massachusetts Institute of Technology. Retrieved from http://hdl.handle.net/1721.1/75005

Harel, I. E., & Papert, S. E. (1991). *Constructionism*. Ablex Publishing.

Hemmendinger, D. (2010). A plea for modesty. *Acm Inroads*, *1*(2), 4-7. doi: 10.1145/1805724.1805725

Henry, C. (2011). *Genre Evolution in Video Games and a Framework for Analysis* (Doctoral dissertation, University of Alberta). Retrieved from http://hdl.handle.net/10048/2178

IBM. (2014). SPSS Statistics (version 21.0.0.0) [computer software]. Available from http://www-01.ibm.com/software/analytics/spss/

IGN. (2014). *Game Reviews*. Retrieved from ca.ign.com/games/reviews?filter=latest

International Society for Technology in Education. (2007). *National Educational Technology Standards for Students* (2nd ed.) Retrieved from: http://www.iste.org/standards/standards-for-computer-science-educators

International Society for Technology in Education. (2011). *Operational Definition of Computational Thinking for K–12 Education*. Retrieved from http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2

Irish Computer Society. (2014). *Scratch teacher training*. Retrieved from http://www.scratch.ie/content/scratch-teacher-training

Kafai, Y. & Resnick, M. (1996) Introduction. In K.B. Kafai & M. Resnick (eds.), *Constructionism in Practice, Designing, Thinking, and Learning in a Digital World*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kelleher, C. (2007). Storytelling Alice (version 2) [computer software]. Available from http://www.alice.org/kelleher/storytelling/

Knuth, D. E. (1985). Algorithmic thinking and mathematical thinking. *American Mathematical Monthly*, p 170-181. doi: 10.2307/2322871

Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning. In *Visual Languages and Human-Centric Computing (VL/HCC), 2010 IEEE Symposium on* (pp. 59-66). IEEE. doi: 10.1109/vlhcc.2010.17

Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, *50*(4), 36-42. doi: 10.1145/1232743.1232745

Lee, M., Ko, Y., Song, H., Kwon, K., Lee, H., Nam, M., & Jung, I. K. (2007). Characteristics of internet use in relation to game genre in Korean adolescents. *CyberPsychology & Behavior*, *10*(2), 278-285. doi: 10.1089/cpb.2006.9958

Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. In *ACM SIGCSE Bulletin* (Vol. 41, No. 1, pp. 260-264). ACM. doi: 10.1145/1539024.1508959

Lucas, K. & Sherry, J. (2004). Sex differences in video game play: A communication-based explanation. *Communication Research*. (October 2004), 31 (5), p. 499-523. doi: 10.1177/0093650204267930

Mayer, R. E. (2011). Multimedia learning and games. In S. Tobias & J.D. Fletcher (eds.), *Computer games and instruction* (p. 281-305). Charlotte, NC: Information Age Publishing, Inc.. doi: 10.1017/cbo9781139547369.037

Metacritic. (n.d.). *Metacritic*. Retrieved October 21, 2014 from http://www.metacritic.com/browse/games/score/metascore/all/all

Microsoft. (2014). Kodu Game Lab (Version 1.4.59.0) [computer software]. Available from http://www.kodugamelab.com/

MIT. (2009). Scratch (Version 1.4) [computer software]. Available from http://scratch.mit.edu/

National Research Council (NRC): Committee on Information Technology Literacy. (1999). *Being fluent with information technology*, National Academy Press. Retrieved from: http://www.nap.edu/html/beingfluent/

Norman, D. A. (2002). *The design of everyday things*. Basic books.

Papert, S. (1971). A Computer Laboratory for Elementary Schools. *Computers and Automation*. Retrieved from http://dspace.mit.edu/handle/1721.1/5834

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S. (1991). Situating Constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 5-23). Norwood, NJ:Ablex.

Perlis, A. J. (1962). The computer in the university. *Computers and the World of the Future*, 180-219.

Piaget, J. (1954). The construction of reality in the child. *New York: Basic*. doi: 10.1037/11168-000

PITAC. (2005). *Report to the President: Computational Science: Ensuring America's Competitiveness*. Retrieved from: http://www.sandia.gov/tecs/studies/CompScienceNITRD.pdf

Rajaravivarma, R. (2005). A games-based approach for teaching the introductory programming course. *ACM SIGCSE Bulletin*, *37*(4), 98-102. doi: 10.1145/1113847.1113886

Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education* (p. 265-269). ACM. doi: 10.1145/1734263.1734357

Riley, D. D., & Hunt, K. A. (2014). *Computational Thinking for the Modern Problem Solver*. CRC Press.

Rollings, A., & Adams, E. (2003). *Andrew Rollings and Ernest Adams on game design*. New Riders.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351-380. doi: 10.1007/s10639-012-9240-x

Tai, R. H., Liu, C. Q., Maltese, A. V., & Fan, X. (2006). Planning early for careers in science. *Science, Vol 312*, no. 5777, p. 1143-1144. doi: 10.1126/science.1128690

Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., & Ni, L. (2013, March). Accelerating K-12 computational thinking using scaffolding, staging, and abstraction. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 609-614). ACM. doi: 10.1145/2445196.2445374

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). A model curriculum for K–12 computer science. *Final Report of the ACM K-12 Task Force Curriculum Committee, CSTA*. doi: 10.1145/792548.611912

University of Alberta Department of Computing Science. (2014). *The iverson computing science exam*. Retrieved from http://launch.cs.ualberta.ca/iverson-exam

University of Alberta Office of the Registrar. (2014). *Faculty of science admission requirements*. Retrieved from http://www.registrar.ualberta.ca/calendar/Admission/Requirements-by-Faculty/15.15.html

Ventura, M., Shute, V. & Jeon, K. (2012). Video game play, personality and academic performance. *Computers & Education*. May 2012; 58 (4), p. 1260-1266. doi: 10.1016/j.compedu.2011.11.022

Whitton, N. (2014). *Digital Games and Learning: Research and Theory*. London, UK: Routledge. doi: 10.4324/9780203095935

Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi: 10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725. doi: 10.1098/rsta.2008.0118

Wing, J. M. (2011). Research notebook: Computational thinking─What and why? *The Link Magazine*, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from http://www.cs.cmu.edu/link

Wolf, M. J. (2008). *The video game explosion*. Greenwood press, 88 post road west, Westport: Greenwood Publishing Group,Inc.

Appendix

Frequency Charts for Evaluations on QPGDS

Table 2.A1

Problem Solving Opportunities (0-5): category frequency totals and percentages.

Criteria	Frequency	Percent
0	1	2.5
1	2	5.0
2	5	12.5
3	14	35.0
4	18	45.0
Total	40	100.0

Table 2.A2

Game contains puzzles (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No puzzles present.	2	5.0
1 Point: Offers simple puzzles; I.e. remembering, understanding, applying	34	85.0
2 Points: Has complex puzzles requiring effort to solve; I.e. analyzing, evaluating, creating	4	10.0
Total	40	100.0

Table 2.A3

Forgiveness (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Cost of failure is high. E.g. No checkpoints or lives system are implemented	10	25.0
1 Point: Cost of failure is low. I.e. Some system exists so the player does not need to start from the beginning. E.g., lives or checkpoints.	30	75.0
Total	40	100.0

Table 2.A4

Goals (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: There are no goals or objectives present in the game.	2	5.0
1 Point: There are goals or objectives present in the game.	38	95.0
Total	40	100.0

Table 2.A5

Challenge incrementally increases as the game progresses (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Levels do not harder,	24	60.0

enemies do not get faster or stronger, etc. as the game progresses.		
1 Point: Levels get harder, enemies get faster or stronger, etc. as the game progresses.	16	40.0
Total	40	100.0

Table 2.A6

Computational Thinking (0-16): category frequency totals and percentages.

Criteria	Frequency	Percent
4	2	5.0
5	1	2.5
6	2	5.0
7	5	12.5
8	3	7.5
9	8	20.0
10	11	27.5
11	3	7.5
12	5	12.5
Total	40	100.0

Table 2.A7

Modularization of game elements to a specific object (0-4): criterion frequency totals and percentages.

Criteria	Frequency	Percent
1 Point: One of sound, variables and code elements pertaining to an object appear on the appropriate objects	5	12.5
2 Points: Two of sound, variables and code elements pertaining to an object appear on the appropriate objects	25	62.5
3 Points: Sound, variables and code elements pertaining to an object appear on the appropriate objects	3	7.5
4 Points: The above and variable operations occur on appropriate objects	7	17.5
Total	40	100.0

Table 2.A8

Effective use of procedures/methods/functions (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No subroutines, procedures or methods exist within an object.	3	7.5
1 Point: Some subroutines, procedures or methods exist within an object...	9	22.5
2 Points: All code segments that could be optimally be refactored into subroutines, procedures or methods are ...	28	70.0

Total	40	100.0

Table 2.A9

Recursive/iterative vs. linear structures (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
1 Point: Some linear structures are present where recursive or iterative structures could be used.	18	45.0
2 Points: Wherever possible iterative or recursive structures are used in place of linear methods.	22	55.0
Total	40	100.0

Table 2.A10

Cascading changes driven from one location (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No changes are driven by the same events where appropriate.	3	7.5
1 Point: Some changes are driven by the same events where appropriate.	9	22.5
2 Points: All changes cascading in game are sent from one event where appropriate.	28	70.0
Total	40	100.0

Table 2.A11

Accurate use of mathematical modeling (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No mathematical models are present or models are not effectively demonstrated.	31	77.5
1 Point: If mathematical models are present, but the game does not use effective models to simulate them.	8	20.0
2 Points: Mathematical models are present and the game uses effective models to simulate them.	1	2.5
Total	40	100.0

Table 2.A12

Proper state initialization (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Game state is not properly initialized. Some states are persistent from different game runs causing errors	1	2.5
1 Point: State is correctly initialized where required.	39	97.5
Total	40	100.0

Table **2.**A13

Game play is clean with at most minor programming errors (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Game play suffers greatly from programming error	14	35.0
1 Point: Game play is not hampered by programming error.	26	65.0
Total	40	100.0

Table 2.A14

Events are driven smoothly in parallel (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
1 Points: Some states carry into other states as code is not set to work concurrently with other code.	15	37.5
2 Point: No noticeable errors; code successfully works in parallel.	25	62.5
Total	40	100.0

Table 2.A15

Customization of Player Experience (0-4): category frequency totals and percentages.

Criteria	Frequency	Percent
1	7	17.5
2	18	45.0
3	9	22.5
4	6	15.0
Total	40	100.0

Table 2.A16

Multiple playthroughs yield different experiences (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Multiple playthroughs yield the same experience.	25	62.5
1 Point: Multiple playthroughs yield different experiences.	15	37.5
Total	40	100.0

Table 2.A17

Allows multiple ways through the game based on player choices (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Player is not given opportunities to make choices throughout the game. i.e. linear game	27	67.5
1 Point: Player can make minor choices affecting game.	13	32.5
Total	40	100.0

Table 2.A18

Players can customize the Player Character (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: The PC is not customizable.	35	87.5
1 Point: The PC can be customized.	5	12.5
Total	40	100.0

Table 2.A19

Player Interaction (0-7): category frequency totals and percentages.

Criteria	Frequency	Percent
0	1	2.5
1	9	22.5
2	11	27.5
3	10	25.0
4	6	15.0
5	3	7.5
Total	40	100.0

Table 2.A20

Non-Player Character interactions with other NPCs (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No NPC interactions with other NPCs.	37	92.5
1 Point: NPC interactions with other NPCs.	3	7.5
Total	40	100.0

Table 2.A21

Player interactions with NPCs (0-3): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No PC interactions with NPCs.	13	32.5
1 Point: Interactions with NPCs are linear and scripted throughout the game. ...	21	52.5
2 Points: Interactions with NPCs change throughout the game. ...	5	12.5
3 Points: PC actions affect NPCs reactions to the PC.	1	2.5
Total	40	100.0

Table 2.A22

Information is revealed as the player needs it (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No information is revealed to the player.	12	30.0
1 Point: Information is told explicitly to the player....	24	60.0
2 Points Information is shown implicitly to the player….	4	10.0
Total	40	100.0

Table 2.A23

A reference can be looked at if the user desires (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No help reference is available.	12	30.0
1 Point: Player controls or other hints are accessible over the course of the game.	25	62.5
2 Points: Player controls or other hints are accessible over the course of the game.	3	7.5
Total	40	100.0

Table 2.A24

Player Immersion (0-5): category frequency totals and percentages.

Criteria	Frequency	Percent
0	3	7.5
1	7	17.5
2	10	25.0
3	7	17.5
4	7	17.5
5	6	15.0
Total	40	100.0

Table 2.A25

Characters animation (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Characters are not animated.	21	52.5
1 Point: PC is animated based on player actions (internal)...	12	30.0
2 Points: Environment and NPCs are animated (external) E.g. NPC’s reaction to player’s attacks.	7	17.5
Total	40	100.0

Table 2.A26

Music, sound, and animation (0-3): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Music, sound, and animation are absent from the game.	3	7.5
1 Point: One of music, sound XOR animation is extremely well done.	8	20.0
2 Points: Two of music, sound, and animation are unified.	15	37.5
3 Points: Music, sound, animation and tasks all create a unified experience.	14	35.0
Total	40	100.0

Table 2.A27

Player Motivation (0-8): category frequency totals and percentages.

Criteria	Frequency	Percent
0	7	17.5
1	6	15.0
2	9	22.5
3	7	17.5
4	8	20.0
5	2	5.0
8	1	2.5
Total	40	100.0

Table 2.A28

Player is motivated to complete tasks given to PC (0-3): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No high score functionality or storyline.	16	40.0
1 Point: High score functionality included.	11	27.5
2 Points: Storyline included.	10	25.0
3 Points: Storyline is developed throughout the game and has an ending.	3	7.5
Total	40	100.0

Table 2.A29

Users assume a role in the game, rather than simply playing (0-2): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Players do not take on or create a persona that they develop throughout the game.	22	55.0
1 Point: Players take on or create a persona that is not developed throughout the game.	16	40.0
2 Points: Players take on or create a persona that they develop throughout the game.	2	5.0
Total	40	100.0

Table 2.A30

Achievements present in the game (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: No immediate, meaningful feedback is given to the player upon accomplishing tasks in the game.	36	90.0
1 Point: Achievements are present, but the player isn't alerted when these are accomplished.	1	2.5
2 Points: Immediate, meaningful feedback is given to the player upon accomplishing additional tasks in the game.	3	7.5
Total	40	100.0

Table 2.A31

Rules, goals and objectives are explicit (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: The rules, goals, and objectives the player must complete are NOT explicit...	12	30.0
1 Point: The rules, goals, and objectives the player must complete are explicit...	28	70.0
Total	40	100.0

Table 2.A32

Interface Usability (0-3): category frequency totals and percentages.

Criteria	Frequency	Percent
1	2	5.0
2	4	10.0
3	34	85.0
Total	40	100.0

Table 2.A33

Meaningful feedback (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: The game does not provide feedback to the player when there is a change of state.	5	12.5
1 Point: The game provides feedback to the player when there is a change of state.	35	87.5
Total	40	100.0

Table 2.A34

Uses traditional control conventions (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: When appropriate to use an existing control scheme, this is ignored ...	1	2.5
1 Point: If appropriate, traditional game control conventions are used.	39	97.5
Total	40	100.0

Table 2.A35

Affordance and visibility (0-1): criterion frequency totals and percentages.

Criteria	Frequency	Percent
0 Points: Objects that the player needs to interact with are NOT salient or it is not obvious with what ...	2	5.0
1 Point: Objects the player needs to interact with are salient and it is obvious with what ...	38	95.0
Total	40	100.0

Conclusion

In the near future, computational thinking will impact everyone, in every field, and will be a fundamental skill used by everyone (Wing, 2006). Computational thinking can be taught through the engaging area of game design. Incidentally, game design and game theory are a viable problem space for pre-service teachers to develop quality pedagogical practice and to further engage their own students through focused goals, challenging tasks, clear and compelling standards, protection from adverse consequences for initial failures, affirmation of performance, affiliation with others, novelty and variety, and choice and authenticity (Jones et al., 1994; Schlechty (1997).

What predisposition do pre-service teachers have when incorporating these digital games into the classroom? The first paper first identified and then divided games created by pre-service teachers in *Scratch* (MIT, 2009) into nine distinct genres as found in literature. Next, the relation of gender, genre, and previous experience with using social media and playing video games was explored. Female pre-service teachers did not prefer any one genre of digital game. In contrast, male pre-service teachers preferred to create violent action games over every other type of game genre. It was discovered that males played more hours of games than females in all explored time periods (e.g., elementary, junior high, high school, and university). In addition, both males and females played more hours of video games in junior high and high school than they do in elementary and university. In terms of social media, there were no significant gender differences, but pre-service teachers increased the amount of social media they used over the specified time periods. Pre-service teachers should be aware of potential predispositions they have if they employ game construction for computational thinking and game design skills.

How can the computational thinking and game design skills required by pre-service teachers be assessed? The second paper first identified these skills and then quantitatively assessed them in the pre-service teachers' created game, proposing the *Quality Practices in Game Design Survey* covering: A) Problem solving, measuring how the created games provide critical thinking and problem solving opportunities in-game, which can be transferred to challenges presented to student in class; B) Computational Thinking, which aims to measure competency in computational basics, and may be universally transferable to most problem solving situations; C) Customization of Player Experience, with constructionist motivation, measures how personalized the player experience is, which can be transferred to a the student experience allowing a meaningful individual experience; D) Player Interaction, which measures how interactive and engaging a game is for the player, which can be transferred to creating similar experiences in a classroom; E) Player Immersion, measures the unified experience in the game, which could also increase engagement in class activities by applying similar attention to detail; F) Player Motivation, measuring the incentive players have to play the game, which can similarly encourage engagement in class materials; and G) Interface Usability, measuring the ease-of-use for interface and game elements, which could be important for all tools and situations teachers provide their students.

Developing computational thinking and game design skills needs to be started early in K-12 to develop into proficiency and cannot be introduced in post-secondary. Before K-12 students can be taught these practical skills, their teachers must first gain a mastery of them. In 2005, Bill Gates stated at the National Educational Summit on High Schools that, "[t]raining the workforce of tomorrow with the high schools of today is like trying to teach kids about today's computers on a 50-year-old mainframe. It's the wrong tool for the times" (p. 1). Today's pre-service teachers, and eventually their students, must acquire strong technological fluencies to prepare themselves for their future in the Information Age (Denton, 2012).

References

Denton, D. W. (2012). Enhancing Instruction through Constructivism, Cooperative Learning, and Cloud Computing. *TechTrends*, 56(4), 34-41. doi: 10.1007/s11528-012-0585-1

Gates, B. (2005, February 26). Prepared remarks. *National Education Summit on High Schools*. Retrieved from http://www.gatesfoundation.org/speeches-commentary/Pages/bill-gates-2005-national-education-summit.aspx

Jones, B., Valdez, G., Norakowski, J., & Rasmussen, C. (1994). Designing learning and technology for educational reform. *North Central Regional Educational Laboratory*.

MIT. (2009). Scratch (Version 1.4) [computer software]. Available from http://scratch.mit.edu/

Schlechty, P. C. (1997). *Inventing better schools: An action plan for educational reform*. San Francisco, CA: Jossey-Bass. doi: 10.1177/019263659708159120

Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi: 10.1145/1118178.1118215

References

Adams, E. (2007). *Fundamentals of game design*. Upper Saddle River N.J.: Pearson Prentice Hall.

Alberta Education. (2014). *Programs of study*. Retrieved from http://education.alberta.ca/teachers/program.aspx

AllGame. (2010). *Game Genres.* Retrieved from http://www.allgame.com/genres.php

Anderson, A. F., & Bavelier, D. (2011). Action game play as a tool to enhance perception, attention and cognition. *Computer games and instruction*, 307-329.

Annetta, L., Mangrum, J., Holmes, S., Collazo, K., & Cheng, M. T. (2009). Bridging Realty to Virtual Reality: Investigating gender effect and student engagement on learning through video game play in an elementary school classroom. *International Journal of Science Education*, *31*(8), 1091-1113. doi: 10.1080/09500690801968656

Anton, G., Harris, S., Ochsner, A., Rothschild, M., & Squire, K. (2013). Patterns of play: Understanding computational thinking through game design. In *Proceedings of the Games, Learning, and Society Conference* (Vol. 3). Retrieved from http://press.etc.cmu.edu/files/GLS9.0-Proceedings-2013-web.pdf

Apperley, T. (2006). Genre and game studies: Toward a critical approach to video game genres. *Simulation Gaming* 2006 37: 6. doi: 10.1177/1046878105282278

Ater-Kranov, A., Bryant, R., Orr, G., Wallace, S., & Zhang, M. (2010). Developing a community definition and teaching modules for computational thinking: accomplishments and challenges. In *Proceedings of the 2010 ACM conference on Information technology education* (pp. 143-148). ACM. doi: 10.1145/1867651

Baird, J. R., Fensham, P. J., Gunstone, R. F., & White, R. T. (1991). The importance of reflection in improving science teaching and learning. *Journal of research in Science Teaching*, 28(2), 163-182. doi: 10.1002/tea.3660280207

Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *Learning & Leading with Technology*, *38*(6), 20-23. Retrieved from http://www.learningandleading-digital.com/learningandleading

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54. doi: 10.1145/1929887.1929905

Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010, June). Using scalable game design to teach computer science from middle school to graduate school. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 224-228). ACM. doi: 10.1145/1822090.1822154

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011, March). Recognizing computational thinking patterns. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 245-250). ACM. doi: 10.1145/1953163.1953241

BELIEVE. (2014). Scriptease (Version 2.8) [computer software] Available from http://webdocs.cs.ualberta.ca/~script/

Berland, M., & Lee, V. R. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, *1*(2), 65. doi: 10.4018/ijgbl.2011040105

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145-157. doi: 10.1016/j.compedu.2013.10.020

Bloom, B. S. (1956). Taxonomy of educational objectives. Vol. 1: Cognitive domain. *New York: McKay*.

Blum, L., & Cortina, T. J. (2007, March). CS4HS: an outreach program for high school CS teachers. In *ACM SIGCSE Bulletin* (Vol. 39, No. 1, pp. 19-23). ACM. doi: 10.1145/1227504.1227320

Boechler, P., Artym, C., DeJong, E., Carbonaro, M., & Stroulia, E. (2014). Computational thinking, code complexity and prior experience in a videogame-building assignment. *Proceedings of the IEEE International Conference on Advanced Learning Technologies,* Athens, Greece. doi: 10.1109/ICALT.2014.118

Boechler, P., Carbonaro, M., deJong, E., Stroulia, E. & Gutiérrez, L. (2011). Video games for teaching and learning: game-building as a teaching tool. In M. Koehler & P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2011* (pp. 2054-2061). Chesapeake, VA: AACE. Retrieved June 25, 2014 from http://www.editlib.org/p/36606

Boechler, P. M., Leenaars, L., & Levner, I. (2008). Recreational vs. educational computer experience: Predicting explicit and implicit learning outcomes in a website search. Las Vegas: Proceedings of the Society for Information Technology & Teacher Education International Conference.

Bolkan, J. (2014, Jan 15). Report: most teachers use social media, worry it will cause conflict. *The Journal*. Retrieved from http://thejournal.com/articles/2014/01/15/report-most-teachers-use-social-media-worry-it-will-cause-conflict.aspx

Bos, B., Wilder, L., Cook, M., & O'Donnell, R. (2014). Learning mathematics through Minecraft. *Teaching Children Mathematics*, *21*(1), 56-58. doi: 10.5951/teacchilmath.21.1.0056

Bourgonjon, J., Valcke, M., Soetaert, R., & Schellens, T. (2010). Students' perceptions about the use of video games in the classroom. *Computers & Education*, 54(4), 1145-1156. doi: 10.1016/j.compedu.2009.10.022

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association,* Vancouver, Canada. Retrieved from http://www.aera.net/

Bruckman, A., and Resnick, M. (1995). *The MediaMOO Project: Constructionism and Professional Community. Convergence*, 1(1): 94-109. doi: 10.1177/135485659500100110

California Legislative Information. (2014). *Bill search*. Retrieved from https://leginfo.legislature.ca.gov/faces/billSearchClient.xhtml

Carbonaro, M. (1997). Making technology an integral part of teaching: The development of a constructionist multimedia course for teacher education. *Journal of Technology and Teacher Education*, *5*(4), 255-280. Retrieved from http://dl.acm.org/citation.cfm?id=274219

Carbonaro, M., Cutumisu, M., Duff, H., Gillis, S., Onuczko, C., Siegel, J., ... & Waugh, K. (2008). Interactive story authoring: A viable form of creative expression for the classroom. *Computers & Education*, 51(2), 687-707. doi: 10.1016/j.compedu.2007.07.007

Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: A gender-neutral attractor to computing science. *Computers & Education*, 55(3), 1098-1111. doi: 10.1016/j.compedu.2010.05.007

Chambers, J. & Carbonaro, M. (2003). Designing, developing, and implementing a course on lego robotics for technology teacher education. *Journal of Technology and Teacher Education*, 11(2), 209-241. Norfolk, VA: AACE. Retrieved from http://site.aace.org/pubs/jtate/default.html

Chou, C., & Tsai, M. J. (2007). Gender differences in Taiwan high school students' computer game playing. *Computers in Human Behavior*, *23*(1), 812-824. doi: 10.1016/j.chb.2004.11.011

Clark, R. E., Yates, K., Early, S., & Moulton, K. (2011). An analysis of the failure of electronic media and discovery-based learning. In K. H. Silber & W. R. Foshay (Eds.), *Handbook of improving performance in the workplace* (pp. 263–297). San Francisco, CA: Pfeiffer. doi: 10.1002/9780470587089.ch8

Code. (2014). *Leaders and trend-setters all agree on one thing*. Retrieved from http://code.org/quotes

Cohoon, J. M. (2006). Just get over it or get on with it: retaining women in undergraduate computing. In J. M. Cohoon, & W. Aspray (Eds.), *Women and information technology: Research on underrepresentation* (pp. 205–237). Cambridge MA: MIT Press. doi: 10.7551/mitpress/9780262033459.003.0007

Columbia Journalism School. (February 10, 2014). *Computational thinking with jeannette wing* [Video file]. Retrieved from https://www.youtube.com/watch?v=U67utvZai8s

Connolly, T.M., Boyle, E. A., Stansfield, M.H., & Hainey T. (2007). The potential of online games as a collaborative learning environment. *International Journal for Advanced Technology for Learning*, 4(4), 208-0923. Retrieved from http://www.actapress.com/Content_Of_Journal.aspx?JournalID=63

Denner, J., Werner, L., & Ortiz, E. (2011). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. *Computers & Education*, 58(1), 240-249. doi: 10.1016/j.compedu.2011.08.006

Denning, P. J. (2009). The profession of IT beyond computational thinking. *Communications of the ACM*, *52*(6), 28-30. doi: 10.1145/1516046.1516054

Denton, D. W. (2012). Enhancing Instruction through Constructivism, Cooperative Learning, and Cloud Computing. *TechTrends*, 56(4), 34-41. doi: 10.1007/s11528-012-0585-1

Dickey, M. D. (2005). Engaging by design: How engagement strategies in popular computer and video games can inform instructional design. *Educational Technology Research and Development*, 53(2), 67–83. doi: 10.1007/bf02504866

Drummond, J. A., & Dubow, E. F. (2011). Violent video games, aggressive behavior, and social/ school adjustment. Retrieved from http://www.education.com/reference/article/violent-video-games-social-school-aggressive/

Elliott, L., Ream, G., McGinsky, E., & Dunlap, E. (2012). The contribution of game genre and other use patterns to problem video game play among adult video gamers. *International journal of mental health and addiction*, 10(6), 948-969. doi: 10.1007/s11469-012-9391-4

ESA, the. (2013). *Demographic and usage data essential facts about the computer and video game industry*. Retrieved from http://theesa.ca/wp-content/uploads/2013/10/Essential-Facts-English.pdf

ESA, the. (2014). 2014 *Essential facts about the computer and video game industry*. Retrieved from http://www.theesa.com/wp-content/uploads/2014/11/ESA-Media-Resources.zip

Extra Credits. (January 22, 2014a). *Extra credits - the magic circle - how games transport us to new worlds*. [Video file]. Retrieved from http://youtu.be/qZ-EY9gTsgU

Extra Credits. (January 29, 2014b). *Extra credits - affordances - how design teaches us without words*. [Video file]. Retrieved from http://youtu.be/QCSXEKHL6fc

Federici, S. & Stern, L. (2011). A constructionist approach to computer science. In S. Barton et al. (Eds.), *Proceedings of Global Learn 2011* (pp. 1352-1361). AACE. Retrieved from http://www.aace.org/

Fischer, K. W., & Immordino-Yang, M. H. (2002). Cognitive development and education: From dynamic general structure to specific learning and teaching. Traditions of scholarship in education. *Chicago: Spencer Foundation*.

Ford, J. L. (2008). *Scratch Programming for Teens*. Course Technology PTR, Boston MA.

Franklin, T. & Annetta, L. (2011). Preface to Special Issue: Digital games and simulations in teacher preparation. *Journal of Technology and Teacher Education*, 19(3), 239-242. Retrieved from http://site.aace.org/pubs/jtate/default.html

Furber, S. (2012). Shut down or restart? The way forward for computing in UK schools. *The Royal Society, London*. Retrieved from https://royalsociety.org/education/policy/computing-in-schools/report/

GameFAQs. (2014). *Category Search*. Retrieved from http://www.gamefaqs.com/search/index.html?game=genre

GameFly, Inc. (2003, February 28). *Genre Definitions*. Retrieved from http://www.mobygames.com/glossary/genres/

Games for Entertainment and Learning Lab. (2014, July 29). *Games for entertainment and learning lab*. Retrieved December 4, 2014, from http://www.gel.msu.edu/

Gates, B. (2005, February 26). Prepared remarks. *National Education Summit on High Schools*. Retrieved from http://www.gatesfoundation.org/speeches-commentary/Pages/bill-gates-2005-national-education-summit.aspx

Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York, NY: Palgrave Macmillan. doi: 10.1145/950566.950595

Gee, J. P. (2005). Good video games and good learning. In *Phi Kappa Phi Forum* (Vol. 85, No. 2, p. 33). THE HONOR SOCIETY OF PHI KAPPA PHI. Retrieved from http://www.phikappaphi.org/publications-resources/

Gee, J. P. (2006). Why game studies now? Video games: A new art form. *Games and culture*, *1*(1), 58-61. Retrieved from http://www.jamespaulgee.com/sites/default/files/pub/GamesCulture-1-1.pdf

Gee, J. P. (2007). *Good video games+ good learning: Collected essays on video games, learning, and literacy*. New York: P. Lang. doi: 10.1177/1555412005281788

Gee, J. P. (2013). *Good video games and good learning: Collected essays on video games, learning and literacy* (2nd ed.). New York, NY: Peter Lang Publishing Inc.

Gee, J.P. (2013). Video games and learning [Video file].

Gershenfeld, A. (2011). Leveling up from player to designer: engaging and empowering youth through making video games. *Knowledge Quest*, *40*(1), 54-59. Retrieved from http://www.editlib.org/p/111220/

Good, J. (2011). Learners at the wheel: Novice programming environments come of age. *International Journal of People-Oriented Programming (IJPOP)*, *1*(1), 1-24. doi: 10.4018/ijpop.2011010101

Google. (2010). *Exploring Computational Thinking*. Retrieved from http://www.google.com/edu/computational-thinking/

Greenberg, B. S., Sherry, J., Lachlan, K., Lucas, K., & Holmstrom, A. (2010). Orientations to video games among gender and age groups. *Simulation & Gaming*, 41(2), 238-259. doi: 10.1177/1046878108319930

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. doi: 10.3102/0013189x12463051

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25-27. doi: 10.1145/1378704.1378713

Hainey, T., Boyle, E., Connolly, T., & Stansfield, M. (2011, June). Gender Differences in Motivations for Playing Computer Games: A Combined Analysis of Three Studies. In *Proceedings of the 7th European Conference on Management Leadership and Governance: ECGBL 2011* (p. 211). Academic Conferences Limited.

Hamlen, K. R. (2010). Re-examining gender differences in video game play: Time spent and feelings of success. *Journal of Educational Computing Research*, 43(3), 293-308. doi: 10.2190/ec.43.3.b

Hamlen, K. R.(2011). Children's choices and strategies in video games. *Computers in Human Behavior*. January 2011, 27 (1), p. 532-539. doi: 10.1016/j.chb.2010.10.001

Harel, I. (1988). *Software design for learning: children's construction of meaning for fractions and LOGO programming*. Massachusetts Institute of Technology. Retrieved from http://hdl.handle.net/1721.1/75005

Harel, I. E., & Papert, S. E. (1991). *Constructionism*. Ablex Publishing.

Hayes, E. (2008). Game content creation and it proficiency: An exploratory study. *Computers & Education*, 51(1), 97-108. doi: 10.1016/j.compedu.2007.04.002

Hayes, E. (2011). Gender and gaming. In J. D. Fletcher & S. Tobias (Eds.), *Computer games and instruction* (p. 453-476). IAP.

Hayes, E., & Ohrnberger, M. (2013). The gamer generation teaches school: The gaming practices and attitudes towards technology of pre-service teachers. *Journal of Technology and Teacher Education*, *21*(2), 154-177. Retrieved from http://www.editlib.org/p/41242/

Heeter, C., Egidio, R., Mishra, P., Winn, B., & Winn, J. (2008). Alien Games Do Girls Prefer Games Designed by Girls?. *Games and Culture*, 4(1), 74-100. doi: 10.1177/1555412008325481

Hemmendinger, D. (2010). A plea for modesty. *Acm Inroads*, *1*(2), 4-7. doi: 10.1145/1805724.1805725

Henry, C. (2011). *Genre Evolution in Video Games and a Framework for Analysis* (Doctoral dissertation, University of Alberta). Retrieved from http://hdl.handle.net/10048/2178

IBM. (2014). SPSS Statistics (version 21.0.0.0) [computer software]. Available from http://www-01.ibm.com/software/analytics/spss/

IGN. (2014). *Game Reviews*. Retrieved from ca.ign.com/games/reviews?filter=latest

International Society for Technology in Education. (2007). *National Educational Technology Standards for Students* (2nd ed.) Retrieved from: http://www.iste.org/standards/standards-for-computer-science-educators

International Society for Technology in Education. (2011). *Operational Definition of Computational Thinking for K–12 Education*. Retrieved from http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2

Irish Computer Society. (2014). *Scratch teacher training*. Retrieved from http://www.scratch.ie/content/scratch-teacher-training

Jenkins, H., & Cassell, J. (2009). From Quake Grrls to Desperate Housewives: A decade of gender and computer games. *Beyond Barbie and Mortal Kombat: New perspectives on gender and gaming*, 5-20. doi: 10.5860/choice.46-6619

Jones, B., Valdez, G., Norakowski, J., & Rasmussen, C. (1994). Designing learning and technology for educational reform. *North Central Regional Educational Laboratory*.

Kafai, Y. & Resnick, M. (1996) Introduction. In K.B. Kafai & M. Resnick (eds.), *Constructionism in Practice, Designing, Thinking, and Learning in a Digital World*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kafai, Y. B. (1996). Learning design by making games: Children's development of design strategies in the creation of a complex computational artifact. In Y. Kafai &M. Resnick (Eds.), *Constructionism in practice: Designing, thinking and learning in a digital world* (pp. 71–96). Mahwah, NJ: Lawrence Erlbaum Associates. doi: 10.4324/9780203053492

Kafai, Y. B., Franke, M. L., Ching, C. C., & Shih, J. C. (1998). Game design as an interactive learning environment for fostering students' and teachers' mathematical inquiry. *International Journal of Computers for Mathematical Learning*, 3(2), 149-184. doi: 10.1023/a:1009777905226

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. *Business horizons*, 53(1), 59-68. doi: 10.1016/j.bushor.2009.09.003

Kebritchi, M., & Hirumi, A. (2008). Examining the pedagogical foundations of modern educational computer games. *Computers & Education*, 51(4), 1729-1743. doi: 10.1016/j.compedu.2008.05.004

Kelleher, C. (2007). Storytelling Alice (version 2) [computer software]. Available from http://www.alice.org/kelleher/storytelling/

Kenny, R. F., & McDaniel, R. (2011). The role teachers' expectations and value assessments of video games play in their adopting and integrating them into their classrooms. *British Journal of Educational Technology*, 42(2), 197-213. doi: 10.1111/j.1467-8535.2009.01007.x

Klopfer, E., Osterweil, S., Groff, J., & Haas, J. (2009). Using the technology of today, in the classroom today. The instructional power of digital games, social networking, simulations, and how teachers can leverage them. *The Education Arcade*, 1-21. Retrieved from http://education.mit.edu/

Knuth, D. E. (1985). Algorithmic thinking and mathematical thinking. *American Mathematical Monthly*, p 170-181. doi: 10.2307/2322871

Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning. In *Visual Languages and Human-Centric Computing (VL/HCC), 2010 IEEE Symposium on* (pp. 59-66). IEEE. doi: 10.1109/vlhcc.2010.17

Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, *50*(4), 36-42. doi: 10.1145/1232743.1232745

Lee, M., Ko, Y., Song, H., Kwon, K., Lee, H., Nam, M., & Jung, I. K. (2007). Characteristics of internet use in relation to game genre in Korean adolescents. *CyberPsychology & Behavior*, *10*(2), 278-285. doi: 10.1089/cpb.2006.9958

Lenhart, A., Jones, S., & Macgill, A.R. (2008a). *Adults and video games*. Washington, DC: Pew Research Center. Retrieved from http://www.pewinternet.org/2008/12/07/adults-and-video-games/

Lenhart, A., Kahne, J., Middaugh, E., Macgill, A. R., Evans, C., & Vitak, J. (2008b). *Teens, video games, and civics: teens' gaming experiences are diverse and include significant social interaction and civic engagement*. Pew Internet & American Life Project.

Lim, C. P. (2008). Spirit of the game; empowering students as designers in schools. *British Journal of Educational Technology*. doi: 10.1111/j.1467-8535.2008.00823_1.x

Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. In *ACM SIGCSE Bulletin* (Vol. 41, No. 1, pp. 260-264). ACM. doi: 10.1145/1539024.1508959

Lucas, K. & Sherry, J. (2004). Sex differences in video game play: A communication-based explanation. *Communication Research*. (October 2004), 31 (5), p. 499-523. doi: 10.1177/0093650204267930

Mayer, R. E. (2011). Multimedia learning and games. In S. Tobias & J.D. Fletcher (eds.), *Computer games and instruction* (p. 281-305). Charlotte, NC: Information Age Publishing, Inc.. doi: 10.1017/cbo9781139547369.037

Mayer, R.E. (2014). *Computer games for learning: An evidence-based approach*. Cambridge, MA: The MIT Press.

Metacritic. (n.d.). *Metacritic*. Retrieved October 21, 2014 from http://www.metacritic.com/browse/games/score/metascore/all/all

Microsoft. (2014). Kodu Game Lab (Version 1.4.59.0) [computer software]. Available from http://www.kodugamelab.com/

Millstone, J. (2012, May). Teacher attitudes about digital games in the classroom. In *The Joan Ganz Cooney Center at Sesame Workshop*.

Mishra, P., & Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *The Teachers College Record*, 108(6), 1017-1054. doi: 10.1111/j.1467-9620.2006.00684.x

MIT. (2009). Scratch (Version 1.4) [computer software]. Available from http://scratch.mit.edu/

Mumtaz, S. (2000). Factors affecting teachers' use of information and communications technology: a review of the literature. *Journal of information technology for teacher education*, 9(3), 319-342. doi: 10.1080/14759390000200096

National Research Council (NRC): Committee on Information Technology Literacy. (1999). *Being fluent with information technology*, National Academy Press. Retrieved from: http://www.nap.edu/html/beingfluent/

Norman, D. A. (2002). *The design of everyday things*. Basic books.

Papert, S. (1971). A Computer Laboratory for Elementary Schools. *Computers and Automation*. Retrieved from http://dspace.mit.edu/handle/1721.1/5834

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S. (1991). Situating Constructionism. In I. Harel & S.Papert (Eds.), *Constructionism* (pp. 5-23). Norwood, NJ:Ablex.

Papert, S. (1998). Does easy do it? Children, games, and learning. *Game Developer*, 5(6), 88. Retrieved from http://www.gdcvault.com/gdmag

Penner, D. E. (2000). Cognition, computers, and synthetic science: Building knowledge and meaning through modeling. *Review of research in education*, 1-35. doi: 10.3102/0091732x025001001

Perlis, A. J. (1962). The computer in the university. *Computers and the World of the Future*, 180-219.

Phan, M. H., Jardina, J. R., Hoyle, S., & Chaparro, B. S. (2012, September). Examining the role of gender in video game usage, preference, and behavior. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 56, No. 1, pp. 1496-1500). SAGE Publications. doi: 10.1177/1071181312561297

Piaget, J. (1954). The construction of reality in the child. *New York: Basic*. doi: 10.1037/11168-000

Piaget, J. (1970). *Genetic epistemology*. Trans. E. Duckworth. doi: 10.1177/000276427001300320

Pierson, M. E. (2001). Technology integration practice as a function of pedagogical expertise. *Journal of Research on Computing in Education*, *33*(4), 413-430. doi: 10.1080/08886504.2001.10782325

PITAC. (2005). *Report to the President: Computational Science: Ensuring America's Competitiveness*. Retrieved from: http://www.sandia.gov/tecs/studies/CompScienceNITRD.pdf

Prensky, M. (2008). Students as designers and creators of educational computer games: Who else?. *British Journal of Educational Technology*, 39(6), 1004-1019. doi: 10.1111/j.1467-8535.2008.00823_2.x

Rajaravivarma, R. (2005). A games-based approach for teaching the introductory programming course. *ACM SIGCSE Bulletin*, *37*(4), 98-102. doi: 10.1145/1113847.1113886

Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education* (p. 265-269). ACM. doi: 10.1145/1734263.1734357

Rideout, V. J., Foehr, U. G., & Roberts, D. F. (2010). *Generation M2: Media in the lives of 8–18 year-olds*. Kaiser Family Foundation. January 2010. <http://www.kff.org/entmedia/upload/8010.pdf> Retrieved 10.05.11.

Riley, D. D., & Hunt, K. A. (2014). *Computational Thinking for the Modern Problem Solver*. CRC Press.

Robertson, J. (2012). Making games in the classroom: Benefits and gender concerns. *Computers & Education*, 59(2), 385-398. doi: 10.1016/j.compedu.2011.12.020

Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computers & Education*, 50(2), 559-578. doi: 10.1016/j.compedu.2007.09.020

Rollings, A., & Adams, E. (2003). *Andrew Rollings and Ernest Adams on game design*. New Riders.

Salen, K. (2007). Gaming literacies: A game design study in action. *Journal of Educational Multimedia and Hypermedia*, 16(3), 301-322. Retrieved from http://www.editlib.org/p/24374

Salen, K. (2011). *Quest to learn: Developing the school for digital kids*. MIT Press.

Salerno, K. (2013). Video Games and Learning: Teaching and Participatory Culture in the Digital Age, by Kurt Squire. *Alberta Journal of Educational Research*, 59(1), 129-132. Retrieved from http://www.ajer.ca/

Schlechty, P. C. (1997). *Inventing better schools: An action plan for educational reform*. San Francisco, CA: Jossey-Bass. doi: 10.1177/019263659708159120

Schmid, R.F., Bernard, R.M., Borokhovski, E., Tamim, R.M., Abrami, P.C., Surkes, M.A., Wade, C.A., & Woods, J. (2014). The effects of technology use in postsecondary education: A meta-analysis of classroom applications. *Computers & Education*, *72*, 271-291. doi: 10.1016/j.compedu.2013.11.002

Schrader, P. G., Archambault, L. M., & Oh-Young, C. (2011). Training by Gaming: Preparing Teachers of Today for Tomorrow's Learning Environments. *Journal of Technology and Teacher Education*, 19(3), 261-286. Retrieved from http://www.editlib.org/p/35993/

Schrader, P. G., Zheng, D., & Young, M. (2006). Teachers' perceptions of video games: MMOGs and the future of preservice teacher education. *Innovate: Journal of Online Education*, 2(3), 1-10. Retrieved from http://www.nyu.edu/classes/keefer/waoe/waoej.html

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351-380. doi: 10.1007/s10639-012-9240-x

Shapiro, J., SalenTekinbas, K., Schwarts, K., & Darvasi, P. (2014). *The MindShift Guide to Digital Games and Learning* [PDF]. Available from http://www.kqed.org/assets/pdf/news/MindShift-GuidetoDigitalGamesandLearning.pdf

Smith, C.P., & Neumann, M.D. (2014). Scratch it out! Enhancing geometrical understanding. *Teaching Children Mathematics*, *21*(3), 185-187. doi: 10.5951/teacchilmath.21.3.0185

Soomro, K. A., Kale, U., & Yousuf Zai, S. (2014). Pre-service teachers' and teacher-educators' experiences and attitudes toward using social networking sites for collaborative learning. *Educational Media International*, (ahead-of-print), 1-17. doi: 10.1080/09523987.2014.977003

Steinkuehler, C., & Squire, K. (2014). *Video Games and Learning*. Retrieved from: https://www.coursera.org/course/videogameslearning

Tai, R. H., Liu, C. Q., Maltese, A. V., & Fan, X. (2006). Planning early for careers in science. *Science, Vol 312*, no. 5777, p. 1143-1144. doi: 10.1126/science.1128690

Tamin, R.M., Bernard, R.M., Borokhovski, E., Abrami, P.C., & Schmid, R.F. (2011). What forty years of saya about the impact of the technology on learning: A second-order meta-analysis and validation study, *Review of Educational Research*, *81*(1), 4-28. doi: 10.3102/0034654310393361

Tobias, S., & Fletcher, J. D. (Eds.). (2011). *Computer games and instruction*. IAP. doi: 10.1111/bjet.12045_2

Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., & Ni, L. (2013, March). Accelerating K-12 computational thinking using scaffolding, staging, and abstraction. In *Proceeding of* *the 44th ACM technical symposium on Computer science education* (pp. 609-614). ACM. doi: 10.1145/2445196.2445374

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). A model curriculum for K–12 computer science. *Final Report of the ACM K-12 Task Force Curriculum Committee, CSTA*. doi: 10.1145/792548.611912

University of Alberta Department of Computing Science. (2014). *The iverson computing science exam*. Retrieved from http://launch.cs.ualberta.ca/iverson-exam

University of Alberta Office of the Registrar. (2014). *Faculty of science admission requirements*. Retrieved from http://www.registrar.ualberta.ca/calendar/Admission/Requirements-by-Faculty/15.15.html

Ventura, M., Shute, V. & Jeon, K. (2012). Video game play, personality and academic performance. *Computers & Education*. May 2012; 58 (4), p. 1260-1266. doi: 10.1016/j.compedu.2011.11.022

Vos, N., van der Meijden, H., & Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education*, 56(1), 127-137. doi: 10.1016/j.compedu.2010.08.013

Whitton, N. (2014). *Digital Games and Learning: Research and Theory*. London, UK: Routledge. doi: 10.4324/9780203095935

Williams, M.K., Foulger, T.S. & Wetzel, K. (2009). Preparing pre-service teachers for 21st century classrooms: Transforming attitudes and behaviors about innovative technology. *Journal of Technology and Teacher Education*, 17(3), 393-418. Retrieved from http://www.editlib.org/p/28216/

Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi: 10.1145/1118178.1118215

Wing, J. (2009). Proceedings from Computer Science Education Conference. Paris, France. doi: 10.1109/iccse.2009.5228576

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725. doi: 10.1098/rsta.2008.0118

Wing, J. M. (2011). Research notebook: Computational thinking─What and why? *The Link Magazine*, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from http://www.cs.cmu.edu/link

Wolf, M. J. (2008). *The video game explosion*. Greenwood press, 88 post road west, Westport: Greenwood Publishing Group,Inc.

Young, M. F., Slota, S., Cutter, A. B., Jalette, G., Mullin, G., Lai, B., ... & Yukhymenko, M. (2012). Our Princess Is in Another Castle A Review of Trends in Serious Gaming for Education. *Review of educational research*, *82*(1), 61-89. doi: 10.3102/0034654312436980