

Building a Serious Game to Teach Secure Coding in Introductory Programming Courses

Nicoletta Adamo-Villani¹, Marcus Oania¹, David Whittinghill¹, Jacob Brown¹ and Stephen Cooper²

¹Purdue University, USA

²Stanford University, USA

Abstract

We report the development and initial evaluation of a serious game that, in conjunction with appropriately designed matching laboratory exercises, can be used to teach secure coding and Information Assurance (IA) concepts across a range of introductory computing courses. The IA Game is a role-playing serious game (RPG) in which the student travels through seven computer techno-inspired environments (IA concept rooms); in each environment he/she learns a different IA concept. After playing each level, the student completes a related CS educational module comprised of a theory lesson and a lab assignment. The game is being created with a user-centered iterative approach that includes two forms of evaluation: formative and summative. In this paper we report the findings of an initial formative evaluation of the first 2 game levels with a group of undergraduate students.

Categories and Subject Descriptors (according to ACM CCS): J.0 [Computer Applications]: General—

1. Introduction

Reports of software-related security breaches are a weekly occurrence in the news. While certainly not all security breaches are programming-related, the scope of the problem is serious. The National Security Agency (NSA) has a set of standards allowing US universities to be designated as centers for academic excellence in information assurance education (CAE/IAE) [NSA11]. Unfortunately, by the end of 2010, only 3% of all institutes of higher education in the US had achieved this designation, most of which focus their Information Assurance (IA) instruction at the graduate level.

There is clearly a tremendous need for computing graduates with a background in IA. However, it is not at all clear that there are many students pursuing PhDs with a focus in IA with those that do complete their PhDs being generally hired by certain government agencies. Thus while we believe that programs such as NSF's Federal Cyber Service: Scholarships for Service [NSF11] and the NSA's Information Assurance Scholarship Program [DoD11] are helpful for building capacity among institutions, efforts need to be made to reach a wider computing audience.

Given that the vast majority of practicing programmers and software engineers do not have advanced degrees, there is also a clear need to give beginning programming students

at the undergraduate level early exposure to secure coding practices and information assurance fundamentals. Curricular recommendations coming from the ACM support this view. The CS2008 document identifies "computer security" (their term for IA) to be one of the three major new focal areas of computing (along with concurrency and net-centric computing) [ACM11]. The work reported in this paper is in response to the need to effectively teach secure coding to beginning programming students.

1.1. Secure coding

Secure coding can be defined as writing code without bugs or vulnerabilities. While it is likely impossible to do in practice, it turns out that many of the most dangerous vulnerabilities result from a reasonably small set of programming errors. And, courses in secure coding tend to focus on making students aware of these common vulnerabilities, and how to write programs without introducing such vulnerabilities into one's code. There are several outstanding textbooks on secure coding (such as [Eri08], [GW03], [HL03], [Sea05], [Sea08], and [VM03]). However, all of the textbooks are targeted at the advanced undergraduate or graduate level. The intent is to "un-teach" students the bad habits they have previously learned, and to get students to think more seriously

about both the correctness of their code as well as its potential vulnerabilities.

In looking through several secure coding texts, the obvious question arose as to why students should not be taught "correct means" to program in the first place, rather than allowing them to learn to program with errors, and then trying to change their bad habits later. One of the authors had a series of e-mail exchanges with several of the secure coding textbook authors regarding the idea of introducing novices to secure coding, and specifically which topics to cover:

Robert Seacord: "I think this [teaching beginners about secure coding] is a great idea... It would help a lot if they [novices] were just taught the correct way to write code, and that code examples they were exposed to did not contain vulnerabilities or illustrate poor coding principles..."

David LeBlanc: "I think this is a great list [of IA topics to teach to novices]... We teach algorithmic complexity, various topics on reliability, but security is also just part of getting a quality product. There's a need to address what should just be part of everyone's day to day job".

While it would be ideal if we could simply contact all of the CS1 textbook authors and make sure that they focus on secure coding in their books and examples, Seacord's comment about an inability to convince a professor to change his own vulnerable code example led us to believe that this would not be a reasonable solution. Instead, an approach is needed that is appealing to both students and instructors, and could help to teach beginning programming students secure coding principles. To meet this need, we are creating a serious game that, in conjunction with appropriately designed matching laboratory exercises, could be used to teach secure coding across a range of introductory computing courses.

In general, CS courses draw examples and assignments from a context or problem domain [GS02], [GS03]. The choice of context can influence both students' motivation [Wil91] and the quality of their learning as it relates to transfer to other domains [DBP99], [KCC*]. A context that students relate to and find relevant can lead to deeper learning, and material that is learned deeply is more likely to transfer to new contexts [Bru93]. Currently, the typical contexts for introductory computing include business applications (e.g., a bank account example for simple classes in object-oriented programming), systems building (e.g., writing functions to format numbers appropriately into strings), and purely abstract problems. We believe that changing the context of introductory courses without changing the skills or concepts introduced will have a dramatic impact on student motivation. We use the context of security in which we create several laboratory exercises, and believe that this context, when augmented with the student playing a serious game, will help the student not only to master the particular computing concept, but also to begin to develop good habits towards secure programming.

1.2. Serious games

Serious gaming repurposes the concepts of videogames and videogame technologies that have been used for commercial entertainment, and uses the gaming approach for training, education, advertising, national defense, general productivity, and more.

A critical consideration in the development of any video game, serious or otherwise, is whether the finished game is fun, or at the very least, compelling. This factor is what makes a serious game a game rather than a simulation or an interactive lesson. It is the "fun" factor that distinguishes serious games from other pedagogical approaches in that the learner is compelled to learn not necessarily due to the subject matter's intrinsic appeal, but rather due to the entertainment value of the gaming activity with which the subject matter is associated. The player of a serious game is motivated to play the game, and in so doing continues the lesson, much longer and with greater attention than he or she would using traditional learning techniques.

Some research has taken place on serious games from a "new media" perspective [DK06], from psychological perspectives [Ree07], [GKH07], as well as from sociological perspectives [SMML08]. As far as learning, empirical evidence to support the assumption that serious games are educationally effective is still limited. Serious games may have positive impacts as education tools [SJB07], but relatively little is known about the cognition of the users who play them [SJ03]. Blunt [Blu09] demonstrated that students taking specially-designed business, economics, and management courses that had an additional serious game component performed better than students that took the course without the serious game component. Wong [WSN*07] found in an NSF study that non-science major students taught physiology using a serious game methodology performed better than those taught using traditional static text and images. Coller et al. [CS09a] [CS09b] taught a numerical methods course centered completely on a serious game in which students wrote programs that would race a virtual car around a track. They discovered that students in this course spent more time out of class working on the material, demonstrated deeper learning, greater intellectual intensity, more intrinsic motivation, and increased engagement as compared to traditional approaches. Further, these students reported greater levels of challenge and concentration as well as greater interest and enjoyment in the subject matter. Rosas et al. [RNC03] found that the use of games in the classroom, via portable handheld devices, led to improved motivation and learning for math and reading in primary school. In a study performed by Papastergiou [Pap09], results showed that a computer game based learning approach was more effective in fostering students' knowledge of computer memory concepts and also provided more motivation for the students versus those who participated in the non-gaming group.

Cooper et al. [CDP03] incorporated Alice game environment into a course for programmers with little or no programming background. Initial observations of the group who had the course with Alice showed a strong sense of design, an appreciation for trial and error, a background for objects, classes, and object oriented programming, amongst other positive reactions. Results of the study showed that students who took the Alice-based course performed better in a Computer Science level one course (CS1) than the total group tested, and significantly better than the students with similar backgrounds, who did not take the Alice course.

Although some studies suggest that serious games can improve learning, there is still no scientific evidence. Thus, there is a need to investigate the role and benefits of serious games in the classroom. The long-term goal of the work reported in this paper is to advance the knowledge in the field by evaluating the effectiveness of serious games for teaching computer science concepts to undergraduate students.

2. The game

The goal of the game is for students taking an introductory computer science course to learn basic IA concepts and develop a solid understanding of proper secure coding, while having an enjoyable experience. The IA Game is a role-playing serious game (RPG) in which the student plays as the main protagonist—Data Detector (DD) (figure 1). DD travels through computer techno-inspired environments (IA concept rooms) and in each environment he/she learns a different IA concept. After playing one level, the student completes the related CS educational module. Each module includes a theory lesson and one or several lab exercises.



Figure 1: Three concept sketches of the Data Detector character:

Gameplay: The game includes seven levels; each level is an engaging IA room in which the player is exposed to a specific IA concept. Upon selecting a particular level, the student is presented with an introduction of the IA concept by the room robot through narration, text and images. The room robot explains the security-related coding issue to the player, how it is related to the game activity and how to solve it. Figure 2 shows sketches of the seven IA robots. The player then takes part in the activity; the challenge of each activity can

be based on: (1) performance time, (2) error rate, or (3) task completion.

Each game level is a self-contained experience that has specific goals and learning objectives; the player can complete the levels in a non-sequential way. Though each level relates to a fundamental concept in information assurance, the levels are not intended to represent literal analogues to the core concepts; they are constructed instead as engaging visual-spatial representations of the educational concepts. This design means there is not a direct mapping between each level and its accompanying lesson; lessons are not explicit, but implied.

The ultimate project goal is to expose students to fundamental IA concepts in a visual fashion that is entertaining and compelling. It is expected that, through sheer exposure, the concepts will insinuate themselves in the user's mind thus making her or him more receptive and willing to learn.



Figure 2: The seven IA robots - final designs.

IA concepts: We chose the following seven security-related coding issues: Validating User Input; Array Range Checking; Buffer Overflow; Operator Precedence; Rounding Errors; Returning Values and Handling Errors; Numeric Overflow/Underflow. The selection of the IA concepts was based on five criteria. First, each issue represents a common coding practice that has direct or indirect security concerns. Second, each issue is easily understandable and accessible by CS1/CS2 students as well as by advanced high school students. Third, each issue supports a lesson/example/practice/test paradigm for student learning. Fourth, by providing varying levels of difficulty, each issue provides the opportunity for differentiated instruction and learning for a wide range of student abilities. Finally, the essence of each coding issue is suitable to be encapsulated in an engaging and compelling game scenario.

2.1. Technical implementation

The platform for the game is based on Autodesk Maya and Unity3D. We use Maya software to model and texture the virtual environments, props and characters and to animate

their functionality. Interactivity with the 3D components is programmed in C# using the Unity game development platform. The choice of the Unity platform was based on the following considerations:

- Unity has an optimized graphics pipeline that supports interactive rendering of complex animated 3D meshes and advanced lighting and textures even on computers with limited graphics capabilities.
- Unity interfaces seamlessly with major 3D animation tools (i.e. Autodesk Maya and 3D Studio Max) and file formats, and allows for instantaneous import and update of asset files and animations.
- Unity supports a wide range of publishing platforms, including: standalone builds for Mac OS and Windows; web delivery through the Unity Web Player Plug-in (3 MB); Wii and iPhone publishing.

The game is deliverable via web or as an exe or app file, and is being designed to run on hardware and software infrastructure that is already widely deployed in universities. Students will be able to use the game on low-end personal computers (PC/MAC) with low-end graphics cards. Different strategies are being used in order to optimize the game performance. Geometric complexity of the 3D models is kept at a minimum, while retaining visual quality, to ensure client hardware can run the application at interactive rates. Normal maps, a technique for simulating complex geometric detail, is used to add fine detail to objects without adding extra geometry. Level of detail is also employed to find an accurate balance between performance and visual quality on the client machine. Furthermore, light maps are implemented to provide high quality lighting for static geometry without impacting performance.

2.2. Game levels completed so far

Thus far, three levels (and accompanying educational modules) have been created. All of the levels are thematically similar. The player is a sentient character named Data Detector who exists within a vast actualized digital world (a la the movie, Tron) and acts as a guardian of his world protecting it from all threats to its stability and integrity. Game demos can be found at: <http://www2.tech.purdue.edu/cgt/i3/IAgame/website/>.

Level 1 The IA concept. The information assurance concept being presented is data validation. Data that enters the system must be of a particular type and size. The execution of invalid data is a threat to system stability and data integrity and thus must be prevented from occurring. Note, these data are not necessarily malicious, but incompatible and thus a source of increased entropy within the current running program.

Gameplay. The player is the inspector of an assembly line in which data is continuously flowing into the main processor. These data are represented by contiguous squares con-

figured as a 2x10 block of squares. We refer to these as "half-blocks". Each half-block is missing certain squares such that it is not a solid 2x10 block - imagine a human smile with missing teeth. An inverse half-block is required to complement the other half-block thus allowing the two half-blocks to merge and become a whole 4x10 block with no missing squares. A screenshot of level 1 is shown in figure 3.

Half blocks flow continually into the system from left to right. It is the job of the player to pair incoming half-blocks with a complementary half-block before the block reaches the CPU on the right side of the screen. The player has a "stamper" tool that contains a number of different half-block combinations. It is the job of the player to position the stamper above the incoming half-block and stamp it with its correct complement before it is loaded into the CPU and executed. If a complement is not found and an un-validated half-block makes it into the CPU, the CPU is damaged and the player's game session is shortened. If too many un-validated half-blocks are processed by the CPU, the game ends.

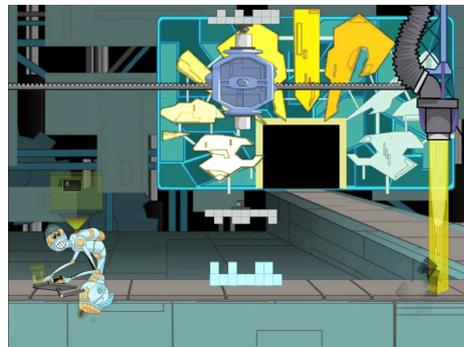


Figure 3: Screenshot of level 1

An additional mechanic the player has available is a laser zapper that can be used as a fallback mechanism to destroy invalid half-blocks. If a complement cannot be found quickly enough, the laser zapper can be employed by the player to vaporize the data. The player must, however, be cautious when using the zapper as the incoming half-block may have had a valid complement available but the player was not able to stamp it in time. In the event this occurs, the CPU is damaged and the game session is shortened.

The game increases in difficulty over time. As more time passes the player is presented with more blocks of greater complexity while also having a larger selection of potential stampers available.

Level 2 The IA concept. The information assurance concept being presented is Returning Values and Handling Errors. Programmers are notorious for not checking return values or handling uncommon errors that are returned from function calls. Security holes are introduced when a malicious user can cause an error to occur that then causes the program to act in an unpredictable or unwanted manner.

Gameplay. In this level, a series of towers in the landscape are exchanging data spheres and sending them to the main system tower (figure 4). The spheres roll down a long, winding track and are then admitted into the system for processing. These spheres contain either valid data or bad data. A chubby, friendly robot represents the former; a bright red spiky monster character represents the latter (figure 5). When the sphere reaches the bottom of the slide, it cracks open and reveals which type of data (character) has gone through.

To thwart potential intrusions, the user has two tools available: an x-ray scanner that allows to see the content inside the spheres, and a paddle with which to smack bad data spheres off the track and into oblivion. All spheres pass through the scanner briefly and reveal themselves (i.e. whether they contain good or bad data). It is up to the player to remember which sphere contains which character and to shove the bad sphere off the track when it approaches.



Figure 4: Screenshot of level 2



Figure 5: Chubby robot representing good data (top); evil spiky creature representing bad data (bottom)

As the game proceeds, the pace of incoming data spheres increases making it more difficult to keep track of which spheres need ejecting. When the player has let 10 bad spheres gain entry, the game ends and has to start over. Conversely, when the player has allowed 10 valid data spheres

into the system the player wins and can proceed to the next level.

Level 3 The IA concept. The information assurance concept being presented is buffer overflow. One commonly used operation in programs is the copying of data into a given area of computer memory. This data may originate from a disk file, user input, a database, the internet, or from another area of computer memory. Before copying the data, the programmer must allocate, or reserve, the correct amount of computer memory. When the programmer does not allocate enough memory or the amount of data copied is unexpectedly larger than the amount of memory allocated, a buffer overflow occurs. Because it is undefined where the overflow data will be written, it is possible for maliciously crafted data to result in unwanted code execution.

Gameplay. The game activity takes place in a bottling plant. The robot in charge of the bottling facility is filling bottles with iced tea. However, the robot is clumsy and not capable of pouring the correct quantity of tea in each bottle. Without the player's intervention, the bottle will be overfilled, will overflow into the adjacent one, and will cause it to destabilize, and explode. The player must place a funnel over the bottle in an attempt to divert enough tea from the stream such that the bottle becomes neither too full nor too empty. Figure 6 shows a screenshot of level 3.

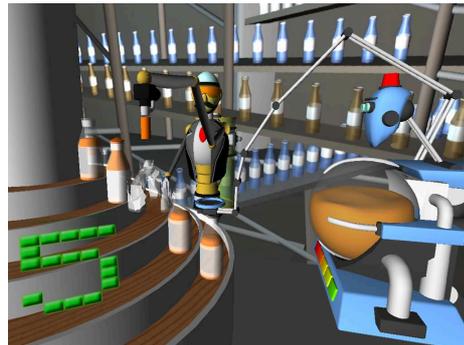


Figure 6: Screenshot of level 3

As the level progresses, more bottles arrive more frequently. For each bottle filled correctly, the player receives a point. For each exploded bottle, the player loses a point. When the player has allowed 10 bottles to explode, she or he will have to start the level again or quit. The player's goal is to obtain the highest score possible before being overwhelmed. The player wins by successfully filling 10 bottles.

3. Initial evaluation

The game is being created using an iterative user-centered development approach that includes two forms of evaluation: formative and summative. Formative evaluation focuses on the design features of the game (i.e., usability, fun

and engagement, user ability to make correlation between the game level and the related educational concept, and quality of the graphics); summative evaluation tests the efficacy of using the serious game and the accompanying educational modules for teaching IA concepts to undergraduate students in introductory programming courses.

In this paper we describe an initial formative evaluation of the first 2 levels of the game with a group of undergraduate students. The goal of the study was to answer the following questions: (1) Is the game usable, engaging and visually pleasing, (2) can students make the correlation between the game level and the related IA concept, and (3) how can the game be improved. In addition, the study also aimed to determine whether gender and video game experience have a significant effect on the ability to play the game.

Study design. The study collected quantitative and qualitative data. Quantitative data included data tracked automatically by the game such as time spent on a particular activity, number of mistakes made while performing a task, number of correct/incorrect answers, and completion/non completion of the game levels. Quantitative data also included students' answers to a web survey with rating questions pertaining to the usability and visual quality of the game levels. Answers were based on a 5 point Likert scale ranging from strongly agree to strongly disagree.

Qualitative data included 'think aloud protocol' and 'critical incidents', i.e. problems encountered that affect task flow and performance, and answers to open-ended questions. In addition, all testing sessions were screen captured and video recorded. The footage was scored with reference to a set of positive and negative instantiations. Positive instantiations that were looked for were smiles, laughing, signs of excitement, and positive vocalization. Negative instantiations were frowns, signs of boredom, signs of frustration, and negative vocalization.

This study was a quasi-experiment, where the sample population was pre-selected. Subjects were recruited randomly from classes and programs at Purdue University as well as Indiana University Purdue University at Indianapolis (IUPUI). Participation in the study was voluntary.

Subjects. Seventeen (17) undergraduate students age 18-23 enrolled in Computer Science, Computer Graphics Technology and Electrical Engineering programs. 4/17 subjects were females.

Procedure. Testing was performed in a controlled lab setting. Each participant was directed to sit in front of a laptop with an external mouse device attached; a video camera was set up near the laptop focusing on the participant's face. Each subject began the test by answering demographic questions via an online survey. Upon completion, the participant was asked to read the online instructions pertaining to the level of the game; then he/she was instructed to open the executable of the game level and play. Participants were allowed

to play the level as long as they pleased. After completion or failure of the level, each subject was asked to refer back to the online survey to answer questions about the level he/she had just played. After all questions for both levels were answered, the participant was asked questions about the overall experience.

Findings for level 1 Game metrics. The mean time spent on level 1 was 193.85 secs. 0/17 subjects were able to complete level 1 on their first attempt; only 1/17 subjects were able to complete the level on his second attempt; 16/17 were unable to complete level 1 but continued played for an average of 3 minutes.

Answers to rating questions. Overall, participants found the level engaging and challenging but difficult to play. When asked: "At any point, I was confused about game play," 15/17 participants agreed or strongly agreed. Participants' reaction to the quality of graphics (i.e. color scheme, look of the environment and robot, and animations) was very positive (mean = 1.8; strongly agree=1 and strongly disagree=5). 13/17 participants were able to make the correlation between the game level and the IA concept of data validation.

Open ended questions. Answers to open ended questions show that subjects did not feel comfortable with the game controls and felt the time allocated to complete certain tasks was not sufficient. 7/17 subjects complained about the game screen being too cluttered and about poor shape visibility. Responses regarding the most fun aspect of the level predominately pointed to the graphics and sound effects. 16/17 subjects responded that they would play the level again.

Observation, video recordings and screen captures. Observation and video recordings of the subjects showed that the majority of the participants were engaged during game play. Initially, participants appeared confused about the game but started to smile and laugh as the level progressed. Although they expressed signs of frustration with the controls (especially the key wheel), they also appeared to be entertained, challenged, and eager to win the level. Screen captures show that participants played the level using a trial and error approach even though they had read the instructions before playing. We believe that this trial and error approach used by the majority of the subjects had a significant impact on completion time and error rate.

Effect of gender and video game experience. An important goal of this project is to create a game that is as gender neutral as possible, and that can also be used by students who do not play video games routinely. A two-sample hypothesis test was performed on the metrics to determine if gender and video game experience (high experience/low experience) were statistically significant. The alpha value used was .05. Results show that gender and video game experience did not have a significant effect on the time spent playing the level and on the error rate. However, the only subject

who completed the level was a male with extensive video game experience.

Findings for level 2 Game metrics. The mean time spent on level 1 was 154.81 seconds; all participants passed the level. 13/17 subjects were able to complete level 2 on their first attempt; 4/17 subjects completed the level on their second attempt.

Answers to rating questions. In contrast to level one, the participants found this level to be easy (mean=1.4); 16 found it fun (mean=1.3). Responses regarding the game controls received positive feedback. 16 of the participants felt that the controls were simple and easy to use (mean=1.7) All 17 participants agreed that 3D graphics and animations were interesting (mean=1.3). 14/17 subjects were able to make the correlation between the level and the related IA concept.

Open-ended questions. Although the level was found easy and fun to play, several participants complained about the placement of the x-ray scanner and the paddle in the environment (too far from the foreground, lack of depth cues). Several requests were made to make the level more difficult and to increase difficulty as game play progresses. 15/17 subjects responded that they would play the level again.

Observation, video recordings and screen captures. Observation and video recordings of the subjects showed that the majority of the participants were engaged and interested in the game. Very few subjects showed signs of frustration with the game controls. 5/17 subjects appeared disappointed when they realized they had won the level so quickly and made the comment that the level is too easy.

Effect of gender and video game experience. Results show that gender and video game experience did not have a significant effect on the time spent playing the level, on the time required to complete the level, and on the error rate.

Discussion of results From this study, it can be concluded that the currently developed levels are usable and engaging and the majority of students are able to make a clear correlation between the game levels and the corresponding IA concepts. Several flaws pertaining to the user interface were revealed within the levels; these problems will be fixed in the next iteration of the levels. Both quantitative and qualitative data prove that the participants enjoyed playing the levels and were motivated to complete the game. Subjects' reactions to the quality of the graphics was very positive, and contributed to keeping the players captivated. Participants enjoyed the difficulty of level one, even though only one participant was able to successfully complete the level. While subjects enjoyed the ease of level two (100% success rate), they thought the level was not challenging enough. These results show that participants find more satisfaction in being challenged, rather than in passing the levels. Finally, findings show that the two game levels are attractive and engaging to novices and female players. This is an important factor, as the goal of the project is to reach a wide audience by cre-

ating a game that is also appealing to females and students who are not avid video game players.

4. Conclusion and future work

In this paper we have described the initial design, development and formative evaluation of a serious game whose goal is to augment the teaching of secure coding practices and principles to novice programmers. The game is being created in response to the need to teach students to program more securely. The overall goal of the project is to provide a demonstration that certain topics in information assurance can be taught as or more effectively using serious games and the accompanying laboratory exercises than by traditional methods. Future work will include completion of the seven game levels and assessment of learning outcomes. Summative evaluation will be conducted once the game is completed to: (1) assess the overall worth and effectiveness of the program; (2) draw out key lessons learned from the project; and (3) determine the sustainability, transferability, scalability, and relative importance of the initiative in enhancing students' understanding of secure coding concepts as well as increase students' retention in computer science majors.

Provided that our work is successful, expanding the serious game approach to other information assurance concepts as well as other subject domains seems to be a logical step in which to proceed. Future iterations could also be expanded to provide a mechanism to assess information assurance knowledge beyond the specific examples within the project to a more general sense of the domain. If we are able to show a correlation between this serious game playing and student attitudes and performance in the classes in which it is used, we expect to expand the tasks to other courses, as well as broadening usage in the target courses.

5. Acknowledgements

This research is supported by the National Science Foundation (NSF) - CCLI program, award #0622900 (Aug. 2010 - Dec. 2013).

References

- [ACM11] ACM: Computer science curriculum 2008: An interim revision of cs 2001, 2011. 1
- [Blu09] BLUNT R.: Do serious games work? results from three studies. 2
- [Bru93] BRUER J. T.: *Schools of thought: a science of learning in the classroom*. MIT Press, 1993. 2
- [CDP03] COOPER S., DANN W., PAUSCH R.: Teaching objects first in introductory computer science. *Proc. of the 34th SIGSCE Technical Symposium on Computer Science Education* (2003), 191–195. 3
- [CS09a] COLLIER B. D., SCOTT M. J.: Effectiveness of using a video game to teach a course in mechanical engineering. *Computers & Education* 53, 3 (2009), 900–912. 2

- [CS09b] COLLER B. D., SHERNOFF D. J.: Video game-based education in mechanical engineering: A look at student engagement. *International Journal of Engineering Education* 25, 2 (2009), 308–317. [2](#)
- [DBP99] DONOVAN M. S., BRANSFORD J. D., PELLIGRINO J.: *How people learn: Bridging research and practice*. National Academy Press, 1999. [2](#)
- [DK06] DOVEY J., KENNEDY H.: *Computer Games as New Media (Issues in Culture and Media Studies)*. Open University Press, 2006. [2](#)
- [DoD11] DOD: Ia scholarship program, 2011. [1](#)
- [Eri08] ERICKSON J.: *Hacking: The Art of Exploitation, 2nd Edition*. No Starch Press, 2008. [1](#)
- [GKH07] GREITZER F. L., KUCHAR O. A., HUSTON K.: Cognitive science implications for enhancing training effectiveness in a serious gaming context. *ACM Journal of Educational Resources in Computing* 7, 3 (2007). [2](#)
- [GS02] GUZDIAL M., SOLOWAY E.: Teaching the nintendo generation to program. *Communications of the ACM* 45, 4 (2002), 17–21. [2](#)
- [GS03] GUZDIAL M., SOLOWAY E.: Computer science is more than calculus: The challenge of living up to our potential. *Inroads – The SIGSCI Bulletin* 35, 2 (2003), 5–8. [2](#)
- [GW03] GRAFF M. G., WYK K. R. V.: *Secure Coding: Principles and Practices*. O’Reilly & Associates, 2003. [1](#)
- [HL03] HOWARD M., LEBLANC D.: *Writing Secure Code, 2nd Edition*. Microsoft Press, 2003. [1](#)
- [KCC*] KOLODNER J. L., CAMP P. J., CRISMOND D., FASSE B., GRAY J., ET AL J. H.: Problem-based learning meets case-based reasoning in the middle school science classroom: Putting learning by design (tm) into practice. [2](#)
- [NSA11] NSA: Centers of academic excellence - institutions, 2011. [1](#)
- [NSF11] NSF: Federal cyber service: Scholarship for service, 2011. [1](#)
- [Pap09] PAPASTERGIOU M.: Digital game-based learning in high-school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education* 52, 1 (2009), 1–12. [2](#)
- [Ree07] REESE D. D.: First steps and beyond: Serious games as preparation for future learning I. *Journal of Educational Multimedia and Hypermedia* 16, 3 (2007). Retrieved January 11, 2010 from Education Module. (Document ID: 1314334661). [2](#)
- [RNC03] ROSAS R., NUSSBAUM M., CUMSILLE P.: Beyond nintendo: design and assessment of educational video games for first and second grade students. *Computers & Education* 40, 1 (2003), 71–94. [2](#)
- [Sea05] SEACORD R. C.: *Secure Coding in C and C++*. Addison Wesley Professional, 2005. [1](#)
- [Sea08] SEACORD R. C.: *The CERT C Secure Coding Standard*. Addison Wesley Professional, 2008. [1](#)
- [SJ03] SQUIRE K., JENKINS H.: Harnessing the power of games in education. 5–33. [2](#)
- [SJB07] SUSI T., JOHANNESSEN M., BACKLUND P.: *Serious games - an overview*. School of Humanities and Informatics, University of Skovde, Technical Report HS-IKI-TR-07-001, 2007. [2](#)
- [SMML08] SCHUURMAN D., MOOR K. D., MAREZ L., LOOY J. V.: Fanboys, competitors, escapists and time-killers: A typology based on gamers’ motivations for playing video games. *Proceedings of the 3rd ACM International Conference on Digital Interactive Media in Entertainment and Art* (2008). [2](#)
- [VM03] VIEGGA J., MESSIER M.: *Secure Coding Cookbook™ for C and C++*. O’Reilly and Associates, 2003. [1](#)
- [Wil91] WILENSKY U.: *Abstract meditations on the concrete and concrete implications for mathematics education. In Constructionism*. Ablex, 1991. [2](#)
- [WSN*07] WONG W. L., SHEN C., NOCERA L., CARRIAZO E., TANG F., BUGGA S., NARAYANAN H., WANG H., RITTERFIELD U.: Serious video game effectiveness. *In Proceedings of the International Conference on Advances in Computer Entertainment Technology* (2007). [2](#)