

Effect of GameMaker on Student Attitudes and Perceptions of Instructors

Marguerite Doman

Winthrop University/Department of Computer Science and Qualitative Methods, Rock Hill, 29733, USA
Email: domanm@winthrop.edu

Merry Sleigh

Winthrop University Department of Psychology, Rock Hill, 29733, USA
Email: sleighm@winthrop.edu

Chlotia Garrison

Winthrop University/Department of Computer Science and Qualitative Methods, Rock Hill, 29733, USA
Email: garrisonc@winthrop.edu

Abstract—Computational thinking including the ability to think critically and solve problems provides benefits for every career path. A positive attitude toward computer science can increase the possibility of students selecting courses that increase computational thinking or pursuing computer science (CS) as a major. This research examined the effect of using GameMaker on the attitudes of students toward computer science (CS) and CS instructors in an introductory CS course. The research consisted of an initial study and a two year longitudinal study. The data was collected using student surveys, qualitative student perceptions, and anonymous teaching evaluations. We hypothesized that students who used GameMaker in their class would show improved attitudes toward CS and would evaluate the instructor more favorably. Our research provides evidence that the incorporation of GameMaker into computer science courses may improve students' short-term attitudes toward computer science and both long-term and short-term perceptions of the class instructor.

Index Terms—GameMaker, general education, computational thinking, attitudes towards computer science.

I. INTRODUCTION

Success in today's global economy requires the ability to think critically, solve problems, and use digital technology effectively. Technology will inform the lives of today's college students regardless of their disciplines. In undergraduate curriculum, introductory computer science courses are provided and often required for non-Computer Science (CS) majors. This course is often the only computer related course in their academic career. Yet, these students, studying business, pre-law, biology, dance, art, education, etc, are the next civic and corporate leaders. They are future educators and influential citizens. They will influence how research money will be spent,

how technology fits in with our society and how our society responds to innovation in technology. An appreciation of STEM (Science, Technology, Engineering, and Mathematics) areas, including computer science, may drive more thoughtful and insightful decision-making processes.

The focus of the undergraduate introductory computer science course is to present issues related to the digital society and introduce computer science concepts and computational thinking [1,2]. Computational thinking [3] is a critical thinking problem-solving technique. The technique includes: » representing information through abstraction, » logically structuring and analyzing data » algorithmic thinking » formulating problems in a way that facilitates the use of a computer or computer tools and » generalizing the process to a wide variety of problems. An effective way of presenting this concept is via programming. Therefore introductory CS courses often include a programming component, such as Java, or HTML (Hyper Text Markup Language) etc. One method of introducing programming taps into the popularity of computer gaming. Multiple visual programming tools have been developed to enable students to design and create their own games.

Our study investigated whether game building projects in an introductory CS course affected attitudes of students towards computer science and computer technology. Tapping onto the popularity of computer games, we examined a visual programming tool that enables students to design and create their own simple games. After introducing game construction, we examined the effect on attitudes after a single semester and whether it impacted teaching evaluations. We revisited our sample after 2 years to determine if the affect persisted. Although previous literature argues strongly for the value of game construction as a teaching tool, its effect on college student attitudes toward computer science and the instructor have not been previously investigated.

We employed the application tool GameMaker®, a program designed by Yoyo Games, Ltd. GameMaker

allows entry-level computing or non-programming students to quickly build simple video games and provides seasoned programmers the tools to build sophisticated games. GameMaker is implemented as a visual programming environment (VPE), using a graphical interface requiring drag-and-drop techniques to insert programming logic and concepts. Research has shown VPEs are successful interventions to introduce programming to new students [4]. GameMaker, in particular, has been successfully utilized to teach programming concepts to children, adolescents, adults, and senior citizens [5, 6]. This tool may be especially useful in engaging both genders, as women show a preference for computer games with a creative, versus solely destructive, component [7].

II. BACKGROUND

Colleges have been increasingly teaching programming to non-Computer Science majors. In 1993, Soloway [8] explored the question of whether all students should be taught to program. The majority consensus was that learning programming is beneficial if it is done effectively. Programming creates better application users, helps students adopt future technology, increases students ability to create new things, and students gain new ways of looking at the world [8]. Teaching programming, which involves problem-solving, could help increase the number of women and minorities that complete computer science courses. Research has shown that an interest in problem-solving influences retention in technology courses. Margolis and Fisher [9] found a lack of perceived problem-solving skills is one reason women drop out of computer science courses in college. Studies have found that students in underrepresented groups in CS that are making career decisions care more about problem-solving and less about technology [10].

In addition, computational thinking skills acquired from learning programming, have benefits for all disciplines. Yadav, et. al. [11] showed that when education majors were presented content that exposed them to the concepts of computing, computational thinking and how these ideas could be used in their future career, student attitudes and understanding were changed. The computational thinking module increased the students understanding that computational thinking is more than computing. It also demonstrated that computational thinking can be taught by teaching problem-solving and critical thinking skills to elementary and secondary students. Freudenthal, et. al. [12] found that students taking mathematics courses with embedded computation were more likely to subsequently take computing courses.

Game construction has emerged as one strategy to interest students in computer science from an early age. Navarrete [13] evaluated middle school students' experiences with digital game design and found that the students perceived the exercise to be both enjoyable and educational. Game construction has also been proven to be effective with young adults. Dalal[14] required

students to build computer games in the college classroom, finding that students perceived the activity to be educational and enjoyable. One reason for the positive response may be that the majority of college students either play or have been exposed to videogames [5, 15].

Developing computer games provides benefits to the individual and the discipline. Whitehead [16] demonstrated that a game design course that required delivery of a working game, primarily using GameMaker, could serve as a useful general education course for non-CS majors. The course also acted as a gateway to new majors, with from 1-3 students declaring computer game design as their major following the course. Overmars [6], pointed out that developing computer games involves many aspects past the more obvious computer graphics, human-computer interactions, and software engineering such as distributed programming, security, simulations and artificial intelligence. In addition, other subject matter including liberal arts, social sciences and psychology are involved. According to Overmars [6], GameMaker is a popular tool for users from age 8 to 80, is useful for demonstrating more advanced concepts, and for teaching at all levels, elementary through college.

Multiple studies have found success in introducing computer science concepts using GameMaker. Guimaraes and Murray [5] used GameMaker in a 4-day summer camp for students in grades 6-10 and found they were able to cover a wide range of topics and students were able to achieve a high level of success in developing their own game. Wang and Wu [17] found that students enjoyed learning computer architecture using a game development project. The positive responses from students exceeded those of students learning software architecture using a robot controller. Vos, Meijden, and Denessen [18] found that the motivation and deep learning of elementary school students that built a simple educational game exceeded that of students that only played a similar educational game.

Hoganson [19] suggests that GameMaker offers advantages for teaching a first programming course because of its integrated development environment allowing the student to seamlessly progress into programming using the GameMaker programming language that is similar to Java. Chamillard [7] describes a course for computer science and non-computer science majors that uses multiple 2D and 3D game creation tools. The students must complete four different games that reportedly motivated student creativity. Exit surveys showed that students were more motivated to pursue computer science as a result of taking the course. The instructors taught the course with a goal of better preparing students for the initial courses required in the computer science major. Providing opportunities for students to gain computer skills at an early age could help increase the number of students that might take a future computer course. Seyal, et.al. [20] in a study of non-computing academics found that computer skill affects computer attitude. Carmichael [21] presented a 5-day course to girls in grades 8 and 9 with a goal of motivating the students to pursue computer courses in high school or

college. Carmichael used creating a video game to help girls become familiar with basic computer science concepts and comfortable with the subject. The majority of the course participants said they would like to use GameMaker in the future and that they would be more likely to try computer science in high school as a result of taking the course.

III. EXPERIMENT 1: GAME CONSTRUCTION AND ITS AFFECT ON ATTITUDES OF NON CSMMAJORS TOWARDS COMPUTER SCIENCE

The purpose of our study was to examine experimentally whether the use of GameMaker altered college students' attitudes over the course of a semester and whether the incorporation of GameMaker into the classroom impacted students' teaching evaluations. We made the following hypotheses:

H1: Students who used GameMaker in their Computer Science class would show improved attitudes toward computer science compared to students who did not use GameMaker.

H2: Students would evaluate the instructor more favorably when GameMaker was used to teach computer programming concepts compared to students whose instructor did not use Game Maker.

A. Methodology

The course used for this investigation was an introductory level course titled Introduction to Computers and Information Processing (CSCI 101). The course was one option to meet the "technology requirement" within the General Education program at a public university in the southeast. The course provided an introduction to general computer concepts, topics related to a digital society and an introduction to programming. As this course was the most commonly chosen option to fulfill the technology requirement, the classes included students across colleges and majors.

All classes introduced programming concepts. The control classes used a single substantial HTML project. In the classes receiving the intervention, a GameMaker project and a shortened HTML project were used.

In the control sections using HyperText Markup Language (HTML) as the primary programming project, each student was instructed to create a set of webpages. The assignments required that web pages include simple tables, inserted images, imbedded style tags and simple JavaScript (for example, inclusion of a date). Introduction to the HTML language was accomplished with a set of lectures and examples targeting the assignment criteria.

In the intervention sections using GameMaker as the programming option, students were given a very short presentation of programming. This was followed by a presentation that guided the students through the first GameMaker tutorial. In Gamemaker, objects are created as key components of the game. When the object is created, it may be assigned an image, called a sprite. The

behavior of the object is managed through events and responses to those events. These events include a keyboard press or a collision with another object. Each event requires that the student designs actions for the object in response to that event. For example, the object may move to the right if the right arrow key is pressed or the score may increase/decrease in response to the collision between two objects. With GameMaker application, a student can quickly create a simple game.

The GameMaker tutorial demonstrated how to build a game in which fruit objects move through a room. Players get points by 'clicking' on the fruit. A timer object is used to control the addition of bombs, adding challenge to the game. The content of the tutorial, along with some short instruction on making objects move by keyboard events, provided the student with enough knowledge to create a simple maze game. The students were then assigned the task to design and create their games. Students were allowed to work individually or in groups of three. For the next two meetings, the students were given short lectures on more complicated tasks, for example: "how to animate an object", and lab time to design and work on their own games. During the lecture/lab the instructor was available to assist and answer any questions. In addition to the initial tutorial and lectures, students were encouraged to visit the official GameMaker site and other sources for resource and ideas (example YouTube.com videos referencing GameMaker). This resulted in finished games that varied from mazes to shooting games to platform games. At the end of the semester, students demonstrated their games to the class.

1. The Sample

Participants were drawn from seven sections of the control class and eight sections of the experimental class across three semesters and included students classified as freshman through seniors. Participant demographics was collected at the end of the semesters. Participants were 124 women and 236 men, and 35 who chose not to identify their gender. The majority of participants who self-identified their ethnicity were Caucasian ($n = 151$) and African-American ($n = 87$). One hundred and thirty-eight students declined to provide their ethnicity, a right that was stated to participants on the informed consent form. The remainder had no more than five students in any ethnic category created by the participants. The sample had 202 freshmen, 82 sophomores, 51 juniors, 25 seniors, and 35 participants who did not identify their year in school. Participants represented the Colleges of Arts and Sciences ($n = 145$), Business Administration ($n = 111$), Education ($n = 47$), and Visual and Performing Arts ($n = 27$); Sixty-five students were undeclared or chose not to identify their major with a specific college.

The demographic data for participants in the control condition was compared to that of participants in the experimental condition. There were no differences between the two groups for percentages related to gender, $X^2(1, N = 360) = .055, p > .05, ns$; ethnicity, $X^2(2, N = 257) = 1.24, p > .05, ns$; upper versus lower classmen, X^2

Table 1. Responses to the Computer Science Survey at the Beginning and End of the Semester.

Survey Question	Start of Semester			End of Semester		
	N	Control	Experimental	N	Control	Experimental
Taking computer science courses is a waste of time.	297	3.34 (1.01)	3.59 * (1.01)	392	2.83 (1.14)	3.09 * (1.34)
I don't like people to think I'm smart in computer science.	306	3.49 (.97)	3.93 ** (.76)	389	2.77 (1.13)	3.12 ** (1.29)
Computer science is my worst subject.	294	3.42 (1.01)	3.71 * (1.01)	387	2.67 (1.33)	3.06** (1.44)
I'm not the type to do well in computer programming	317	2.87 (1.08)	3.21 ** (1.11)	392	2.91 (1.24)	3.01 (1.28)
I think that computer science is interesting.	321	3.26 (1.06)	3.36 (1.07)	395	3.35 (1.20)	3.72 ** (1.01)
I can learn to understand computing concepts.	321	4.35 (.75)	4.18 (.86)	395	4.16 (.98)	4.47 ** (.68)
I am confident in my ability to discuss computing/ IT topics	321	2.76 (1.11)	2.54 (1.03)	395	2.80 (1.14)	3.21 ** (1.08)
I think I could handle more advanced work in computer science.	322	2.96 (1.20)	2.96 (1.10)	395	3.03 (1.30)	3.43 ** (1.21)
The challenge of solving problems using computer science appeals to me.	322	2.96 (1.14)	3.14 (1.04)	395	3.19 (1.33)	3.53 ** (1.18)
It is harder to use computer science concepts and programs to solve problems than to try to solve the problems in other ways.	320	2.75 (.87)	2.82 (.97)	394	3.14 (1.16)	2.90 * (1.08)
When I have a problem to solve, I try the first think I think about.	322	2.15 (1.03)	2.29 (1.04)	393	3.46 (1.26)	2.84 ** (1.36)
Knowledge of computing will allow me to secure a better job.	322	4.19 (.89)	4.12 (.93)	394	3.99 (.96)	4.27 * (.86)
Having background knowledge and understanding of computer science is valuable in and of itself.	321	4.06 (.87)	3.94 (.85)	395	3.94 (.93)	4.18 ** (.81)
When I have a problem to solve, I think about the steps I need to do to solve it.	322	4.39 (.78)	4.34 (.71)	395	4.36 (.74)	4.48 (.61)
When I don't understand a programming problem immediately, I feel excited by the challenge.	321	2.48 (1.08)	2.44 (1.04)	394	2.67 (1.25)	2.85 (1.23)
I would rather have someone give me the solution to a difficult programming problem than to have to work it out for myself.	315	2.81 (1.07)	2.73 (1.08)	394	3.14 (1.22)	2.95 (1.25)
My career goals do not require that I learn computing skills.	292	3.44 (1.05)	3.42 (1.17)	392	2.87 (1.25)	2.93 (1.36)
I expect to use computer applications in my future educational and career work.	320	4.04 (.94)	4.14 (.90)	393	3.91 (1.05)	4.08 (1.0)
I see computer science as a subject I will rarely use in my daily life.	309	3.32 (1.04)	3.53 (.98)	394	2.85 (1.16)	2.92 (1.30)
Programming is of no relevance to my life.	312	2.92 (1.03)	3.11 (1.06)	393	2.96 (1.20)	2.93 (1.20)

** $p < .01$ * $p < .05$

(1, $N = 360$) = .172, $p > .05$, ns; and college of major, X^2 (3, $N = 330$) = .86, $p > .05$, ns. Thus, the control and experimental condition were well matched.

2. Evaluation Instruments

Data was collected through three instruments.

Computer Science Attitude Survey. We assessed participants' attitude by using select questions taken from the Computer Science Attitude Survey [22]. The original scale includes 50 statements divided into five sub-sections: confidence in learning computer science and programming, attitude toward success in computer science, computer science as a male domain, usefulness of computer science and programming, and effective motivation in computer science and programming.

Participants respond to positive and negative statements on a 5-point Likert scale

Where 1 represents strongly agree and 5 represents strongly disagree. For the purposes of this study, we did not use the full scale. Twenty questions were selected that targeted attitudes that matched the specific goals of the instructors of the CSCI 101 sections (see Table 1). Three of the 20 questions assessed participants' self-perception of their ability to succeed in computer science (e.g., I'm not the type to do well in computer programming; Computer science is my worst subject). Seven questions assessed participants' perceptions of how applicable computer science and programming are to success in the future (e.g., Knowledge of computing will allow me to secure a better job; I see computer science as a subject I

will rarely use in my daily life). Six questions assessed participants' perceptions of the process of problem solving (e.g., I would rather have someone give me the solution to a difficult programming problem than to have to work it out for myself; When I have a problem to solve, I think about the steps I need to do to solve it). Four questions assessed participants' interest and self-efficacy related to computer science (e.g., I think that computer science is interesting; I am confident in my ability to discuss computing/IT topics). Participants also answered demographic questions that included gender, year in school, ethnicity, and college of major.

Qualitative Student Perceptions of GameMaker. Select sections of the experimental condition were asked the following question at the end of their written final exam in the course: "Describe one thing in this class you found particularly interesting. Why". The response was open-ended. Students were offered additional points to be added to their exam grade if they chose to respond in order to encourage thoughtful answers; these points were awarded regardless of how the question was answered. The question did not specifically apply to their GameMaker experience, but gave the opportunity to volunteer opinions of the experience.

Anonymous Teaching Evaluations. A standardized, department-wide teaching evaluation instrument was used for all sections of CSCI 101. The instrument assessed 19 aspects of the course including the instructor, grading, the syllabus, and the textbook. For the purposes of this study, we focused on eight specific questions that we felt would be most likely to be impacted by the use of GameMaker: The class atmosphere, as created by the instructor, was conducive to learning; The instructor was enthusiastic about teaching; The instructor used relevant examples; The instructor encouraged students to put forth their best effort; The instructor challenged students to think critically; The instructor used class time effectively; This course significantly increased my knowledge about the subject matter; The instructor was an effective teacher. We also assessed a ninth question that was not expected to differentiate the classes: The instructor graded tests and assignments fairly. Students responded on a 5-point Likert scale with 1 representing Strongly Disagree and 5 representing Strongly Agree. A higher rating indicated a more favorable evaluation.

3. Procedure

During fall of 2010, teaching evaluations were collected from multiple sections of the CSCI 101 course. GameMaker was not used during that period of time. Instructors were divided into two groups. One instructor agreed to incorporate GameMaker into the class in future semesters setting up the future experimental condition. Three instructors agreed to continue teaching the class without GameMaker, setting up the future control condition.

The following year, the experimental instructor incorporated GameMaker into multiple sections of the class. Data collection did not begin until 2012. In this way, the experimental and control condition were both

established classes and differences between the groups were unlikely to be due to instructor inexperience or novelty for students.

During fall and spring of 2012, the questionnaire instrument was distributed during class time to the control and experimental sections on the first two days of class and again on a class day during the last two days of class (at the end of the semester). Students in all sections were asked to complete teaching evaluations during class time at the end of the semester. All instruments were completed anonymously in a classroom setting, with no person-identifiable information collected.

In spring 2012, students in all experimental sections also completed the open-ended question on their final exam asking them to describe an interesting element of the course.

B. Results

Computer Science Attitude Survey. The survey was given at the beginning and end of three different semesters. Because no person-identifiable data was collected there was no way to link an individual's answers at the start of the semester with that individual's answers at the end of the semester. In addition, the survey was only distributed to students who were in class. There is no way to assess whether the exact same individuals took the survey at the beginning and end of the semester. The two data sets do not entirely represent independent groups or within-subjects design. For this reason, we examined the beginning of the semester and end of the semester data independently of each other.

We used a 2 (gender) X 2 (condition) MANOVA to analyse the data collected at the beginning of the semester. Results revealed no significant interactions on any of the variables examined. We had many students who did not report their gender, resulting in a smaller overall sample size for the MANOVA. In order to take advantage of all students in the two conditions, regardless of whether they reported gender or not, we ran a One-Way MANOVA comparing the two conditions (control and experimental). Results revealed four differences between the control and experimental conditions. See Table 1. Compared to the control condition, students in the experimental sections began the semester by more strongly agreeing that a) Taking computer science courses is a waste of time, $F(1, 295) = 4.70, p = .031$; b) I'm not the type to do well in computer programming, $F(1, 315) = 7.44, p = .007$; c) Computer science is my worst subject, $F(1, 292) = 6.01, p = .015$; and d) I don't like people to think I'm smart in computer science, $F(1, 304) = 19.48, p < .001$. In other words, students in classes that would become the experimental condition began their semesters with a more negative attitude toward computer science than did students in what would be the control condition.

We used a 2 (gender) X 2 (condition) MANOVA to analyse the data collected at the end of the semester. Results revealed only one significant interaction, $F(1, 324) = 11.66, p = .001$. Men in the control condition were less likely than men in the GameMaker condition and women

in both conditions to agree with the statement, “I can learn to understand computing concepts.”

We then collapsed gender and focused exclusively on the two conditions using a One-Way MANOVA. We were able to take advantage of a larger sample size once we collapsed gender, because many of our participants had chosen not to report their gender and thus had not been included in the 2 X 2 MANOVA. See Table 1. Results revealed that three of the four differences found at the start of the semester were still present. Compared to the control condition, students in the experimental sections ended the semester by more strongly agreeing that a) Taking computer science courses is a waste of time, $F(1, 390) = 3.96, p = .047$; b) Computer science is my worst subject, $F(1, 385) = 7.44, p = .007$, and c) I don't like people to think I'm smart in computer science $F(1, 387) = 7.48, p = .007$. In contrast, the difference between the two groups on the statement, “I'm not the type to do well in computer programming” was no longer significantly different, $F(1, 390) = .615, p = .43, ns$.

The MANOVA also revealed differences between the two conditions that were not present at the outset of the semester. Compared to the control classes, students who used GameMaker were more likely to agree that: I can learn to understand computing concepts, $F(1, 393) = 13.37, p < .001$; I think I could handle more advanced work in computer science, $F(1, 393) = 9.87, p = .002$; I think that computer science is interesting, $F(1, 393) = 10.69, p = .001$; I am confident in my ability to discuss computing/IT topics, $F(1, 393) = 12.98, p < .001$; The challenge of solving problems using computer science appeals to me, $F(1, 393) = 7.12, p = .008$; Knowledge of computing will allow me to secure a better job, $F(1, 392) = 9.42, p = .002$; Having background knowledge and understanding of computer science is valuable in and of itself, $F(1, 393) = 7.67, p = .006$. Compared to the control classes, students who used GameMaker were less likely to agree that: It is harder to use computer science concepts and programs to solve problems than to try to solve the problem in other ways, $F(1, 392) = 4.26, p = .04$, and When I have a problem to solve, I try the first thing I think about, $F(1, 391) = 21.98, p < .001$.

The two conditions did not differ on the following variables: When I have a problem to solve, I think about the steps I need to do to solve it [$F(1, 393) = 3.33, p = .07, ns$]; When I don't understand a programming problem immediately, I feel excited by the challenge [$F(1, 392) = 2.0, p = .16, ns$]; I would rather have someone give me the solution to a difficult programming problem than to have to work it out for myself [$F(1, 392) = 2.25, p = .14, ns$]; My career goals do not require that I learn computing skills [$F(1, 390) = .23, p = .63, ns$]; I expect to use computer applications in my future educational and career work [$F(1, 391) = 2.68, p = .10, ns$]; I see computer science as a subject I will rarely use in my daily life [$F(1, 392) = .35, p = .55, ns$]; Programming is of no relevance to my life [$F(1, 391) = .07, p = .79, ns$].

Qualitative Student Perceptions of GameMaker
Eighty-one students provided a response to the open-ended question on their final exam. Forty-six of these

comments pertained directly to GameMaker. In other words, when asked to comment upon an “interesting” aspect of the course, 57% selected GameMaker as their focus. Two raters evaluated each comment and categorized it as positive, negative, or mixed (representing a combination of positive and negative comments). With an inter-rater reliability of .98, 45 comments were coded as positive and 1 was coded as mixed. Sample comments can be seen in Table 2.

Anonymous Teaching Evaluations. We did not collect demographic data on the teaching evaluations; thus, gender was not evaluated. Seven sections of the control class ($n = 151$) were compared to eight sections of the experimental class ($n = 224$) using a MANOVA. The experimental sections of the course were evaluated more positively than the control sections of the class on the following six variables: increased knowledge about the subject matter [$F(1, 374) = 6.07, p = .014$], fairness of grading [$F(1, 373) = 9.33, p = .002$], conduciveness of the class to learning [$F(1, 371) = 12.70, p < .001$], use of relevant examples [$F(1, 371) = 6.47, p = .011$], instructor encouraging students' best effort [$F(1, 373) = 8.65, p = .003$], and instructor challenging students to think critically [$F(1, 373) = 13.74, p < .001$]. In addition, the experimental sections had higher ratings for the overall evaluation of teaching effectiveness, $F(1, 373) = 10.44, p = .001$. The two groups did not differ on how enthusiastic they perceived the instructor to be [$F(1, 373) = 1.61, p = .21, ns$] and how efficiently the instructor used class time [$F(1, 372) = 2.76, p = .10, ns$]. See Table 3 for means and standard deviations.

In order to assess the possibility that the higher teaching ratings were due to the instructor rather than the inclusion of GameMaker, we compared teaching evaluations for the experimental versus the control instructors during the semester prior to the incorporation of GameMaker. The responses for “The instructor was an effective teacher” between the two established groups were found to be significantly different, $t(134) = 2.32, p = .02$, with the experimental instructor performing significantly worse than the control instructors. The instructor who would become part of the experimental condition had a mean of 3.79 ($SD = 1.40$) across two sections with evaluations completed by 39 students. The instructors who would become part of the control condition had a mean of 4.27 ($SD = .92$) across five sections with evaluations completed by 97 students.

C. Discussion

We hypothesized that students who used GameMaker in their Computer Science class would show improved attitudes toward computer science. This hypothesis was partially supported. The two groups could not be differentiated on sixteen variables at the beginning of the semester; however, by the end of the semester, the GameMaker classes responded to nine of those variables indicating a more favorable attitude toward computer science compared to the control classes. The GameMaker students reported that they felt they could learn about computer science, found computer science interesting,

felt confident discussing it, and considered the information to be useful. The pattern of these results provides strong evidence that the inclusion of GameMaker in the classroom impacted overall attitudes toward computer science. These results are similar to those reported by Kurkovsky [23] who found that students who developed mobile game modules in an introductory CS course showed a higher level of engagement than students who did not develop game modules.

We found only one interaction between condition and gender of participant. Men who were in the control conditions felt least able to understand computing concepts at the end of the semester. In a previous study, Wilson [24] found that computer science students demonstrated a strong preference for application and game-making activities over mathematical problems, with men and CS majors exhibiting the strongest

preference. This previous finding offers one possible explanation as to why men in our study who were unable to use GameMaker reacted more negatively to the class than men who could use GameMaker and more negatively than women.

The overall lack of gender differences in student attitudes may be noteworthy. Past research has consistently revealed that women lack confidence in their technical ability [25, 26]. However, a study by Wilson [27] also found no significant gender difference in predictive factors leading to success in an introductory computer science course. Our study suggests that GameMaker may have similar influences on men and women.

The GameMaker students also demonstrated some change in their problem solving ability. Compared to the control students, the GameMaker students' responses suggested that they did not perceive computer science as

Table 2. Sample Student Comments from the Open-Ended Response Question on the Final Exam

	Sample Student Comments
Mixed Comment	<ul style="list-style-type: none"> I am absolutely fascinated with the technology and it is always nice to learn more about it, I mean. I hated GameMaker, but it's great to know that it exists and what it does.
Positive Comments	<ul style="list-style-type: none"> Learning how to make our game was interesting because I actually got to experience the other side of the spectrum for a change. I'm usually the one playing the game, so to get a chance to be the one making it was definitely a great experience. One of my most favourite topic to learn about was GameMaker because it was interesting to learn exactly how games work and be able to create my own. It was not what I expected. I found the entire process of learning how to use GameMaker interesting. I like to play computer games and through this process I have gained some insight into how the games are made and that was really interesting. The GameMaker project was fun because it included creativity and allowed me to actually program /design something of my own I really found the game interesting. It was amazing to take one of my favourite games, Oregon trails, and create something similar, Winthrop Trail. To be able to play it like a real game has been amazing. It gave me a start to create more games. I am currently making a game for my daughter. I know she will love it I learned how hard it is to make even a simple game ant that was with a program designed to make them! It was fun though. I learned how to create a game. I thought it was pretty interesting. It's not very day you make your own computer game. I was most interested in GameMaker. The program challenged me to apply both what was learned as well as creativity. I learned that designing a game takes a lot of work because it tome several days just (to) make a simple computer game. I have much more respect for the game designers now, because now I know the time and effort put into making a game. I can only imagine the amount of time and the level of difficulty it must take to design a difficult game. I learned how to use a program to create my very own game. I found this interesting because I have never created my own game on the computer before and it was fun. I thought learning to make a game was interesting because I had always wondered how computer games were made. I found learning how to make games interesting. I have always wanted to know who they are made Now that I know how difficult it is to make them, I will appreciate them a lot more. I absolutely LOVED using GameMaker. I am not interested in programming, typically, but because of the visual and easy to use layout, I was hooked. I will be making a game over the summer. One thing that I learned in this class that was particularly interesting was how to make a game and website. As I get older, I plan on having my own personal website. So having some insight on how to do so already is extremely beneficial. I love to play games, so learning how to make a game was interesting and fun. I now can make my own games for fun that will entertain me and be special because I made it. Learning how to create my own game and how to write its functions Before this class I thought you just basically drew out picture for them and used computers to simulate those pictures... thus creating a game. I thought the GameMaker and the HTML sections were particularly interesting. This is because it gave me the basic knowledge I needed as well as refresh my memory on some things I learned in high school. GameMaker however was complete new and exciting to figure out. I learned a ton of things in the class but the one thing that interested me the most was/making the GameMaker game. It was really interesting because being able to say that I created a game and could play it afterward just seems super cool to me!

Table 3. Means and Standard Deviations for the Control and Experimental Teaching Evaluations

	Control N = 151	Experimental N = 224
This course significantly increased my knowledge about the subject matter.	4.01 (1.24)	4.28 (.92)*
The instructor was an effective teacher.	4.06 (1.20)	4.39 (.80)*
The instructor graded tests and assignments fairly.	4.46 (.86)	4.69 (.60)*
The class atmosphere, as created by the instructor, was conducive to learning.	3.99 (1.13)	4.35 (.82)*
The instructor used relevant examples.	4.30 (.96)	4.52 (.75)*
The instructor encouraged students to put forth their best effort.	4.19 (.94)	4.44 (.74)*
The instructor challenged students to think critically.	3.99 (1.05)	4.35 (.82)*
The instructor was enthusiastic about teaching.	4.28 (.90)	4.40 (.80)
The instructor used class time effectively.	4.42 (.73)	4.55 (.68)

* denotes significance at $p < .02$

a difficult way to solve problems, and they were less likely to solve problems by simply doing the first thing that came to mind. This finding is supported by observed outcomes in high school classrooms, where game construction was believed to improve higher-order thinking skills in both boys and girls [28].

Despite these group differences, students in our study responded similarly on seven variables examining attitudes toward computer science. Based on their mean responses, students in general agreed that they think about the steps needed to solve a problem when faced with a problem, agreed that they would rather have the solution to a difficult programming problem given to them rather than having to work it out for themselves, and neither agreed or disagreed that the challenge of programming problems is exciting. We also found that students were in agreement about how frequently they would use computer science in their careers and daily lives. Our sample was drawn from a range of majors across the university campus, reflecting diversity in students' future plans. The majority of students enrolled in the class to meet a General Education requirement, not a requirement for their major. Thus, it may not be surprising that the use of GameMaker did not result in students reporting a greater need for computer science in the future or in altered problem solving strategies.

It is worth noting that four variables differentiated the two groups at the start of the semester. The students who were enrolled in the GameMaker sections had more negative attitudes toward computer science and their ability to do well. For example, they were more likely to agree that taking computer science courses was a waste of time and that they were not the type to do well in computer programming. One possible explanation for the differences found at the start of the semester could be that students with more negative feelings toward computer science may have intentionally self-selected into the GameMaker classes, finding video game creation to be more interesting than the programming activities in the other sections. The instructor had been teaching the class using GameMaker for three semesters prior to data collection; students may have been aware of the class content by talking to their peers. If this were the case,

then the GameMaker instructor not only faced students who had negative attitudes toward computer science but also those who had expectations about the course activities that had not dispelled their negative attitudes.

At the end of the semester, three of these differences remained; however, both the control and experimental groups demonstrated a decrease in these negative attitudes. In addition, the GameMaker students ended with similar scores to the control students on "I'm not the type to do well in computer programming." These findings add weight to the more positive attitudes seen in the GameMaker sections at the end of the semester. Relative to the control students, the GameMaker students began with a more pessimistic attitude toward the class but ended with a pattern that indicated more favorable attitudes toward computer science.

The qualitative data collected from GameMaker students provided additional evidence that perceptions of GameMaker were favorable. When given an opportunity to comment on an interesting aspect of their Computer Science class, the majority of students chose to comment on GameMaker, and the vast majority of those comments were entirely positive. It is possible that students commented positively on GameMaker to please the instructor; however, students were unaware that the incorporation of GameMaker was being systematically evaluated. In addition, students who were motivated to please the instructor in some way had the option of commenting favorably upon the instructor rather than a specific course activity, and this pattern did not emerge in the data. Another possibility is that students commented on GameMaker because it was one of the last class activities, and thus, was on their mind. Recency might explain the high percentage of comments related to GameMaker, but it would not explain the positive nature of the comments. In general, our findings reveal consistent evidence that students responded positively to the incorporation of GameMaker.

We also hypothesized that students would evaluate the instructor more favorably when GameMaker was used to teach computer programming concepts. This hypothesis was supported. On seven of nine variables, the GameMaker students rated the instructor more favorably

than the control students evaluated their instructors. GameMaker students were more likely to perceive their instructor as being fair, creating a classroom environment that facilitated learning, using relevant examples, encouraging and challenging students, and being an effective teacher. Across both conditions, students rated their instructors as similarly enthusiastic and using classroom time effectively. One possible explanation is that the GameMaker instructor had stronger teaching skills than the comparison instructors; however, a comparison of evaluations taken before this study began indicated that the GameMaker instructor received lower teaching ratings than the comparison instructors prior to implementing GameMaker in the classroom. In other words, the inclusion of GameMaker is likely to be the causal effect of the improved teaching ratings.

IV. EXPERIMENT 2: LONGITUDINAL STUDY

The results from Experiment 1 suggest that the inclusion of GameMaker in the computer science classroom changed students' attitudes over the course of the semester. This is a new finding that has not been explored by other research. Experiment 2 was designed to assess whether this attitude change extends beyond the semester in which the students are enrolled in the computer science class. We made the following hypotheses:

H1: Students who used GameMaker in their Computer Science class would have more positive attitudes toward computer science than students not exposed to GameMaker approximately two years following completion of the course.

H2: Students who used GameMaker in their Computer Science class would evaluate the use of Game Maker, the course, and the instructor more positively than would students not exposed to GameMaker approximately two years following completion of the course.

A. Methodology

1. The Sample

Students enrolled in CSCI during Experiment 1 were issued an invitation, using the e-mail contact that they provided to the university in their official records, to participate in Experiment 2. One hundred and nine students responded with a mean age of 21.58 ($sd = 5.02$). We categorized participants in two ways. Each participant had to correctly identify the instructor of their course and whether GameMaker was utilized as a course assignment. Three participants indicated that they did not remember one of these two factors and were thus eliminated from the statistical analysis. The resulting sample was 75 students from the classes that used GameMaker and 31 from the control classes. Of participants that provided their gender, 33 were men and 71 were women; 33 were sophomores, 62 were upperclassmen (juniors and seniors), and 8 reported

“other” which may have indicated that they graduated. Students' self-reported ethnicity was categorized into three groups, resulting in 27 African-American, 70 non-Hispanic Caucasian and 9 other ethnicities. When we compared the demographic characteristics of the participants in the experimental versus control group, they did not significantly differ on age, $t(102) = .09$, $p = .93$, ns, self-reported ethnic categories, $X^2(2, N = 106) = 5.42$, $p > .05$, ns, or class status, $X^2(2, N = 104) = 5.35$, $p > .05$, ns. The experimental group, which had an overall higher response rate, also had a higher percentage of female respondents compared to the control group, $X^2(1, N = 104) = 3.94$, $p < .05$.

2. Evaluation Instruments

Data was collected through two instruments.

Computer Science Attitude Survey. Participants responded to the same 20 questions taken from the Computer Science Attitude Survey that were used in Experiment 1.

Researcher Developed Questions: We also developed a series of 16 questions to assess students' memories of the computer science course. Two questions assessed students' perceptions of how much they learned in the course (I remember a lot of what was covered in that class; I was surprised by how much I learned in CSCI). Two questions assessed students' memories of the teacher (I would recommend the specific teacher that I had to a friend; I think the teacher understood where students were coming from and what their interests were). Three questions assessed how valuable students felt the course information was for their future endeavors (CSCI provided me with helpful skills for the job market; CSCI provided me with skills that helped me with other classes that I have taken since CSCI; The material in CSCI has helped me understand how computing relates to the real world). Three questions assessed whether students remembered the class in a more favorable or unfavorable light (I would recommend the CSCI class to a friend; The class was fun; Overall, I remember the CSCI class in a positive light). These statements were evaluated on a 5-point Likert scale with 1 representing Strongly Disagree and 5 representing Strongly Agree. A higher score indicated more favorable responses. One question assessed how difficult students remembered the class to be using a 5-point Likert-type scale ranging from Very Easy to Very Difficult.

Five additional items were designed to assess students' perceptions of the use of GameMaker as a class activity. A brief description of GameMaker was provided to help GameMaker students recall the activity and control students be aware of the activity in which they did not participate. Sample questions included: Students who used GameMaker probably learned more than those who were taught without it; The classes that used GameMaker were probably more fun than the classes without it. Responses were made on a 5-point Likert scale with 1 representing Strongly Disagree and 5 representing Strongly Agree.

Last, students responded to four demographic

that assessed age, gender, ethnicity, and current school status (freshmen, sophomore, junior, senior, other).

3. Procedure

At the end of the 2014 spring semester, the survey was posted online through Qualtrics [29]. Students who were enrolled in the computer science courses during the semester when Experiment 1 was conducted were invited via e-mail to participate. The e-mail list was generated by Records and Registration and included the official university e-mails of students who were still enrolled as well as students who had graduated; there was no way to assess the number of students who were actively using that particular e-mail account. Students were offered the incentive of being entered in a drawing to receive a \$40 Amazon gift card for participation.

Computer Science Attitude Survey: We compared participants in the control condition to those in the experimental condition using an independent t-test. There were no significant differences between the two groups on their responses to the 20 variables assessed on the Computer Science Attitude Survey.

Researcher Developed Questions: We compared the control and experimental participants using an independent t-test. Means and standard deviations are shown in Table 4. The two groups significantly differed on the two questions related to their memories of the teacher. Compared to the control participants, experimental participants were more likely to agree that they would recommend the specific teacher they had to a friend [$t(104) = 2.12, p = .04$] and that they believed that the teacher understood students and their interests [$t(101) = 2.12, p = .04$]. The two groups also differed on the three questions related to their overall memory of the course. Compared to the control participants, experimental participants were more likely to agree that they would recommend the CSCI class to a friend [$t(104) = 3.18, p = .002$], the class was fun [$t(104) = 2.17, p = .03$], and

that they remembered the class in a positive light [$t(103) = 2.69, p = .008$].

The results related to learning were less consistent. Compared to the control participants, experimental participants were more likely to agree that they were surprised by how much they learned in CSCI, $t(103) = 2.96, p = .004$; however, the two groups did not differ when asked how much they remembered of what was covered in the class, $t(102) = .28, p = .77, ns$. The two groups did not differ on the three questions related to how valuable the course information was for their future endeavours, with p values ranging from .275 to .664, ns , or on their rating of the course difficulty, $t(104) = -.29, p = .78, ns$.

Control and experimental participants differed on all five questions related specifically to the use of GameMaker in CSCI. Compared to the control participants, experimental participants were more likely to agree that GameMaker is a great way to teach students about computer science [$t(103) = 3.39, p = .001$], students who used GameMaker probably learned more than those who were taught without it [$t(102) = 4.29, p < .001$], classes that used GameMaker were more fun than those without GameMaker [$t(102) = 3.52, p = .001$], classes that used GameMaker probably rated the professor more favorably [$t(103) = 3.19, p = .002$], and all CSCI 101 classes should be required to use GameMaker [$t(103) = 2.92, p = .004$].

4. Discussion

We hypothesized that the positive attitudes toward computer science observed in the experimental participants from Experiment 1 would still exist two years after the class. This hypothesis was not supported. The longitudinal data suggests that these outcomes did not persist over time; rather, students who worked with GameMaker reported similar attitudes as students who did not experience GameMaker. In addition, students

Table 4. Means and Standard Deviations for the Control and Experimental Longitudinal Responses

	Control $N = 31$	Experimental $N = 75$
I would recommend the specific teacher that I had to a friend.	3.42 (1.29)	3.93 (1.07)*
I think the teacher understood where students were coming from and what their interests were.	3.07 (1.08)	3.55 (1.04)*
I would recommend the CSCI class to a friend.	2.84 (1.13)	3.59 (1.09)**
The class was fun.	2.42 (1.12)	2.95 (1.15)*
Overall, I remember the CSCI class in a positive light.	3.07 (1.14)	3.69 (1.05)**
I was surprised by how much I learned in CSCI.	2.84 (1.07)	3.54 (1.13)**
I remember a lot of what was covered in the CSCI class.	3.07 (1.08)	3.14 (1.13)
CSCI provided me with skills that helped me with other classes that I have taken since CSCI.	2.84 (1.16)	3.11 (1.14)
The material in CSCI has helped me understand how computing relates to the real world.	3.47 (.73)	3.55 (1.0)
CSCI provided me with helpful skills for the job market.	3.06 (.93)	3.17 (1.12)
In general, how difficult do you remember this class to be?	2.45 (1.0)	2.40 (.77)
GameMaker is a great way to teach students about computer science.	2.80 (.85)	3.59 (1.15)**
Students who used GameMaker probably learned more than those who were taught without it.	2.69 (.89)	3.57 (.96)**
The classes that used GameMaker were probably more fun than the classes without it.	3.0 (1.13)	3.80 (1.0)**
The classes that used GameMaker probably rated the instructor more positively than did classes without it.	2.90 (.92)	3.52 (.89)**
All CSCI classes should be required to use GameMaker.	2.67 (1.16)	3.35 (1.05)**

** $p < .01$ * $p < .05$

reported similar longitudinal perceptions of how valuable computer science information was for other college classes or their future. One possibility for these results is that students' attitudes toward technology and computer science are based on a lifetime of experiences; while they may be temporarily influenced, long-term attitudes might be difficult to change. As an extension of this idea, students were further along their individual career paths two years after taking this introductory computer science class, and that focus may have driven their perceptions of the value of computer science more than did a class taken earlier in their college tenure.

Although the positive attitudes toward Computer Science observed in the experimental condition were not maintained over time, the immediate change in attitude that we found in Experiment 1 may still have value. College students are regularly selecting courses, thinking about their majors, and considering career options. The positive attitudes toward computer science engendered by the experience with GameMaker may have had influenced students' immediate choices related to coursework or their willingness to consider careers requiring computer concepts. Future studies may want to more specifically examine how the short-term attitudes relate to behaviour.

In contrast to the attitude questions about computer science, students who used GameMaker still retained the more favorable impression of the instructor seen in Experiment 1, as well as a more favorable impression of the class overall. Students who used GameMaker also were more likely to agree that they were surprised by how much they had learned. These findings support our second hypothesis and cannot be attributed to the ease or difficulty of the course assignments, because our two groups rated the course similarly on this variable. Adding further support to the likelihood that the differences in perception were related to GameMaker, our two groups revealed consistent differences in opinions when asked specifically about GameMaker itself. Students who had experience with GameMaker believed that it was a good teaching tool and served to improve perceptions of the instructor. Taken together, these data suggest that the use of GameMaker resulted in positive long-term perceptions of the class and teacher.

It might be worth noting that we had a significantly higher response rate from students, especially women, in the GameMaker condition than students in the control condition. This outcome may have been the result of the unpredictability of real-world data collection. However, future researchers may want to further investigate other explanations. One possibility is that the GameMaker students were more willing to respond to a request for information about the class because of their more favorable impressions. Another possibility is that the longitudinal data favoring the GameMaker condition may be biased by the low response rate from the control condition students.

V. GENERAL DISCUSSION

Our study had limitations created by the collection of data in a real world setting. Students were not randomly assigned to the classes, and thus, our comparison groups did not hold identical attitudes at the outset of data collection. Similarly, the instructors for the control versus experimental conditions were not identical. Due to constraints at our university, we were unable to collect person-identifiable information, preventing us from examining how individual attitudes changed from the start to the end of the semester. We documented these factors to the extent possible and incorporated that information into the interpretation of our results.

VI. CONCLUSIONS AND FUTURE WORK

Our research showed that GameMaker, a visual game building tool, positively affected the attitudes of students towards technology and their perceptions of the class and instructor. Engaging students in computer science concepts, particularly computational thinking is a challenge. Yet, a positive attitude toward computer science will benefit those that interact with technology. While most non-majors may not become computer scientists or programmers, their appreciation of and attitude toward the skill will inform their future decisions. Their future decisions may affect future policy or simply adaption of technology. A positive attitude toward technology increases the opportunity for students to develop computational thinking skills.

Our research findings suggest that GameMaker has potential to improve student attitudes not only about computer science but about the instructor teaching the course. A more positive attitude about the instructor can increase the possibility of the instructor successfully teaching more difficult concepts and students' receptivity to mastering such concepts. In addition, a more positive attitude toward computer science may increase students' willingness to take additional computer science courses or consider computer science as a potential major.

Future work may include identifying other tools and the specific aspects about these tools that improve student attitudes. Identifying specific aspects has the potential to inform and enhance new learning environments. The lack of gender differences in student attitudes may be noteworthy and suggests additional research in that direction. In addition, future researchers may want to assess if the short-term positive attitudes toward computer science elicited by the use of GameMaker influenced students' immediate behaviors. For example, it is possible that students' attitudes affected their willingness to enroll in courses with a technology component and/or to consider careers utilizing technology. Last but not least, future researchers may want to investigate instructors' perceptions of GameMaker, as previous research has focused on student perspectives.

In general, our study provides evidence that the incorporation of GameMaker into computer science courses may improve students' short-term attitudes toward computer science and both long-term and short-

term perceptions of the instructor and class. Previous research supports the argument that GameMaker is enjoyable to students [13, 15 and 26]. Our study indicates that engaging in desired classroom activities also has the potential to positively impact students' perceptions of the class and the instructor. This outcome may be particularly meaningful given that these students represented both genders and a variety of college majors.

REFERENCES

- [1] R. Kick, AP CS principles: engaging, challenging, and rewarding, *ACM Inroads*, vol. 6. No. 1, pp. 42-45, March 2015.
- [2] Joint Informatics Europe & ACM Europe Working Group on Informatics Education 2013, Informatics education: Europe cannot afford to miss the boat. <http://europe.acm.org/iereport/ACMandIEreport.pdf>, accessed June 2015.
- [3] J. Wing, (2006). Computational Thinking, *Communications of the ACM*, vol. 49, no 3, March 2006.
- [4] L.A. Siiman, M. Pedaste, E. Tõnisson, R. Sell, T. Jaakkola and D. Alimisis (2014). A Review of Interventions to Recruit and Retain ICT Students. *International Journal of Modern Education and Computer Science*, vol. 6, no. 3 pp.45-54, DOI: 10.5815/ijmecs.2014.03.06.
- [5] M. Guimaraes, &M. Murray, (2008). An exploratory overview of teaching computer game development. *Journal of Computing Sciences in College*, vol. 24, no. 1, pp. 144-149, October 2008.
- [6] M. Overmars, (2004). Teaching computer science through game design. *Computer*, vol. 37, no. 4, pp. 81-83, April 2004.
- [7] A. T. Chamillard, (2006). Introductory game creation: No programming required. SIGCSE, Houston, TX, *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pp. 515-519, March 2006.
- [8] E. Soloway, (2014). "Should we teach students to program?" *Communications of the ACM*, vol. 36, no. 10, October 1993: 21+. Academic OneFile. Web. 6 May 2014. http://0-go.galegroup.com.library.winthrop.edu/ps/retrieve.do?sgHitCountType=None&sort=DA-SORT&inPS=true&prodId=AONE&userGroupName=winthropudl&tabID=T002&searchId=R3&resultListType=RESULT_LIST&contentSegment=&searchType=AdvancedSearchForm¤tPosition=29&contentSet=GALE%7CA14468382&&docId=GALE|A14468382&docType=GALE&role=&docLevel=FULLTEXT.
- [9] J. Margolis, &A. Fisher, (2002). *Unlocking the clubhouse: Women in computing*. Cambridge, MA: The MIT Press, 2002.
- [10] D. Agosto, M. Atwood, and S. Gasson, (2008), Changing mental models of the IT professions: a theoretical framework, *Journal of Information Technology Education*, vol. 7, pp. 205-221, 2008; <http://jite.org/documents/Vol7/JITEv7p205-221Agosto323.pdf>.
- [11] A. Yadav, N. Zhou, C. Mayfield, S. Hambrusch, J. Korb, (2011). Introducing Computational Thinking in Education Courses, *SIGCSE'11*, March 9-12, 2011.
- [12] E. Freudenthal, A. Duval, S. Hug, A. Ogrey, K. Lim, C. Tabor, R. Gonzalez, and A. Siegel, (2011). "Planting the seeds of computational thinking: An introduction to programming suitable for inclusion in STEM curricula" *Proceedings of the 118th American Society for Engineering Education Annual Conference & Exposition*, Vancouver, Canada, June 2011. Available at: http://works.bepress.com/kien_lim/24.
- [13] C.C. Navarrete, (2013) Creative thinking in digital game design and development: A case Study. *Computers & Education*, vol. 69, pp. 320-331.
- [14] N. Dalal, (2012). Teaching tip: Using rapid game prototyping for exploring requirements discovery and modeling. *Journal of Information Systems Education*, vol. 23, no. 4, pp. 341-344.
- [15] M. Terlecki, J. Brown, L. Harner-Steciw, J. Irvin-Hannum, R. Marchetto, L. Ruhl, andJ. Wiggins, (2011). Sex differences and similarities in video game experience, preferences, and self-efficacy: Implications for the gaming industry. *Current Psychology: A Journal for Diverse Perspectives on Diverse Psychological Issues*, vol. 30, no. 1, pp. 22-33.
- [16] J. Whitehead, (2008). Introduction to Game Design in the Large Classroom, *GDCSE '08*, Feb28-Mar 3, 2008, Miami, FL.
- [17] A. Wang and W. Bian (2009). An Application of a Game Development Framework in Higher Education, *International Journal of Computer Games Technology*, vol. 2009, Article ID 693267, pp. 1-12.
- [18] N. Vos, H. Meijden, and E. Denessen (2011), Effects of constructing versus playing an educational game on student motivation and deep learning strategy use, *Computers & Education* vol. 56, no. 1, pp. 127-137, January 2011.
- [19] K. Hoganson, (2010). Teaching programming concepts with GameMaker. *Journal of Computing Sciences in College*, JCSC, vol. 26, no. 2, pp. 181-188, December 2010.
- [20] A. Seyal, R. MahbuburandN. Rahman, (2000), Computer attitudes of non-computing academics: a study of technical colleges in Brunei Darussalam, *Information & Management*, vol. 37, no. 4, pp. 169-180, June 2000.
- [21] Carmichael, G. (2008). Girls, computer science, and games. *SIGCSE Bulletin*, vol. 40, no. 4, pp. 107-110, November 2008.
- [22] E. N. Wiebe, L. Williams, K. Yang, andC. Miller, (2003). Computer Science Attitude Survey. Report No.: NCSU CSC TR-2003-1. Department of Computer Science, NC State University, Raleigh, NC.
- [23] S. Kurkovsky, (2013) Mobile game development: Improving student engagement and motivation in introductory computing courses. *Computer Science Education*, vol. 23, no. 2, pp. 138-157, 2013, <http://dx.doi.org/10.1080/08993408.2013.777236>.
- [24] B. C. Wilson, (2006). Gender differences in types of assignments preferred: Implications for computer science instruction. *Journal of Educational Computing Research*, vol. 34, no. 3, pp. 245-255, 2006.
- [25] A. Settle, J. Lalor, and T. Steinbach. (2015). Reconsidering the Impact of CS1 on Novice Attitudes. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*. ACM, New York, NY, USA, 229-234. DOI=10.1145/2676723.2677235.<http://doi.acm.org/10.1145/2676723.2677235>
- [26] K. Singh, K. R. Allen, R. Scheckler, andL. Darlington, (2007). Women in computer-related majors: A critical synthesis of research and theory from 1994 to 2005. *Review of Educational Research*, vol. 77, pp. 500-533, 2007. doi: 10.3102/0034654307309919.

- [27] B. C. Wilson, (2002). A Study of Factors Promoting Success in Computer Science Including Gender Differences. *Computer Science Education*, vol. 12, no. 1-2, pp. 141-164, 2002.
- [28] M. Carbonaro, D. Szafron, M. Cutumisu, and J. Schaeffer, (2010). Computer-game construction: A gender-neutral attractor to Computing Science. *Computers and Education*, vol. 55, no. 3, pp. 1098-1111, 2010.
- [29] Qualtrics software, Version 1.611s (0.281, 0.224, 0.031, 0.131, 0.019) of the Qualtrics Research Suite. Copyright © 2014 Qualtrics. Qualtrics and all other Qualtrics product or service names are registered trademarks or trademarks of Qualtrics, Provo, UT, USA. <http://www.qualtrics.com>, accessed June 2015.

Authors' Profiles



Marguerite Doman, PhD. Dr Doman is an assistant professor at Winthrop University. She received her doctorate from the University of North Carolina at Charlotte in Computer Science. She worked as a systems developer at IBM for 20 years working on operating systems for midrange computers including S/3, S/36, S/38 and AS/400. Her

research interests include computer networking and computer science education.



Merry Sleight, PhD. Dr. Sleight is a professor in the Department of Psychology at Winthrop. She received her doctorate in developmental psychology from Virginia Tech and did her postdoctoral fellowship at Indiana University. She directed the Psychology Honors Program at George Mason University prior to joining the faculty at Winthrop in 2003. Sleight

regularly teaches "Research Methods," "Biopsychology," "Learning," and "The Human Experience," as well as special topics honors courses. Her research interests focus on neonatal development, and she is heavily involved in supporting undergraduate research. She currently serves as the Psi Chi Southeastern Vice-President and the Associate Editor of the *Journal of Psychological Inquiry*.



Chlotia Garrison, PhD. Dr Garrison is an associate professor in the Department of Computer Science and Qualitative Methods at Winthrop University. She received her doctorate from Florida State University. She worked almost 20 years as a civilian mathematician and software engineer for the Department of Defense at Eglin AFB. Her

teaching areas include Software Engineering, Programming Languages, Introductory Programming and Computer Literacy. She research interests include computer security from the user perspective and promoting computer science.