Motivating High School Students and Teachers to Create Interactive Software: Can Summer Workshops Affect Participants' Interest for Developing Games and Animations?

Eliana C. Medina

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

Doctor of Philosophy

University of Washington

2008

Program Authorized to Offer Degree: College of Education

UMI Number: 3318223

Copyright 2008 by
Medina, Eliana C.

All rights reserved.

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UMI Microform 3318223
Copyright 2008 by ProQuest LLC.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

©Copyright 2008
Eliana C. Medina

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Eliana C. Medina

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

Stephen Kerr

Reading Committee:

Stephen Kerr

John Bransford

Allen Glenn

Date: 6/10/08

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to ProQuest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature Eliana Medina

Date 6/10/08

University of Washington

Abstract

Motivating High School Students and Teachers to Create Interactive Software: Can Summer Workshops Affect Participants' Interest for Developing Games and Animations?

Eliana C. Medina

Chair of the Supervisory Committee:
Professor Stephen Kerr
College of Education

This study examines the appropriateness of animations and games as attractive motivators to encourage high school (HS) students and teachers to apply math and science concepts to interactive software development. To identify factors that could motivate participants to learn about animation and game technology, the researcher observed a group of students and teachers attending a two-week summer camp and a teacher workshop lasting two weeks as well. Comparisons of participants' final projects and feedback from their learning experiences were documented. Two websites were created to support the workshops with additional materials and video tutorials. The purpose of having the websites was to observe if they support the development of interest in creating animations and games.

Data were collected from 12 students and 23 teachers using five surveys aimed at capturing participants' feedback about previous educational experiences and their experiences in the workshops. Additionally, three sets of programming assignments for each participant were gathered, as well as class materials, researcher's field notes, and results from instructors' interviews.

Students and teachers showed no significant changes in their interest for creating games and animation after attending the workshops. Yet, results from the follow-up

survey, two months after the summer institute ended, indicated a significant decline in teachers' interest for creating interactive software. Students made progress faster, struggled less with the content, and handled the technical problems better than HS teachers. In contrast, teachers were more likely to work extra time on their projects during and after the teacher institute compared to the HS students. The class website proved to be more useful for the teachers than for the students, but the online community was used only by a small number of participants.

Teachers found value in the educational programs they created in the teacher institute and they used them in their classrooms. Students were inspired to take mathematics and computer technology classes.

Table of Contents

Page

List of Figures

Figure Number Page

List of Tables

Acknowledgments

I would like to thank my parents, my sister, my aunt, my partner, and the Acker family for their unconditional support and encouragement, without them this effort could not have been possible.

All my advisors and committee members were there for me when I needed them, and influenced my research in many different levels. Bill Winn, John Frederiksen, Steve Kerr, Allen Glenn, John Bransford, Steve Tanimoto, and Barbara Mones taught me importance lessons about teaching and learning, and inspired me through their actions.

The members of the Human Interface Technology Laboratory (HITLab) have been extremely helpful in making my graduate school experience very practical and accepting me as part of the HITLab family. Special thanks to Suzanne Weghorst and Tom Furness for opening the doors of their research lab to me and for their kindly support. Also I would like to thank my good HITLabers friends, Ruth Fruland, Amlan Mukherjee, and Yu-Chien (Nancy) Chen which made my ride through graduate school more pleasant and colorful.

Last, I would like to thank all the instructors and TAs who participated in the summer workshops I did my research on. I appreciate their willingness to allow me to collect my data in their workshops.

Dedication

In memory of my advisor, Bill Winn

Chapter 1: Introduction to the Study

Background of the Problem

Today, the design and development of computer animations and computer games are predominately conducted by male artists, computer scientists and engineers. The gender gap in the gaming industry, however, has attracted more attention from the media, government and academia than the lack of women in the animation industry. As a result, there are more specific data about gender imbalance in the game industry than in the animation industry.

A recent demographic survey on the game industry in North America and the United Kingdom (IGDA, 2005) reveled that only 11.5% of the game developers are female. When it comes to diversity, the study found that 83.3% of the developers are white, while only 2% are African-American, 2.5% are Hispanic, 7.5 % are Asian, and 4.7% are other ethnicities.

Although, there is not precise information about the demographics of the animation industry, it seems like the amount of female animators is substantially lower than the male counterpart. For example, trends on students taking computer animation classes at the college level for the past six years at a Northwest university affirm these gender differences. A rough estimate of women representation provided by a veteran computer animator and professor indicate that the animation industry is composed of about "70% male and 30% female animators." Regarding the number of women and men that usually take her computer animation capstone class, the instructor estimated that the ratio is usually 80% male to 20% female college students. This indicates that the animation industry might be sharing similar challenges to those faced by the game industry.

One reason for the focus on women's underrepresentation in the game industry has to do with the social implications of having young people playing games designed by male game developers for male audiences that reinforce negative stereotypes. Animations, on the other hand, have been historically perceived as appropriate both for male and female audiences, so there has been less concern about who creates animations. However, it is equally important to bring women and minority students into different areas of the computer graphic industry to provide more diverse graphic products and better working environments that support different points of view and are culturally sensitive.

Another reason why researchers are focusing on female and minority underrepresentation in game development may be associated with the fact that game development is usually related to doing programming and solving complex technical problems while computer animation is associated with graphic arts, storytelling and design. While there seems to be a gender and ethnicity balance in the graphics art and design careers, the fact that gender and ethnicity are unbalanced in technology driven careers has been documented extensively (Ogazalek, 1989; Beyer, Rynes, Perrault, Hay & Haller, 2003; Vegso 2007; NCWIT, 2006; Cohoon, & Aspray, 2006; Natale, 2002).

Game development assignments have been used extensively in computer science education for the past ten years with different degrees of success for engaging students with the programming task (Becker, 2001; Kafai, 1994; Sung et al, 2008). For this reason, the potential of game development to motivate diverse students, in particular novice programmers, to actively participate in the creation of interactive software has been studied by educational institutions for many years, and more recently organizations and industry such as the MacArthur Foundation, National Science Foundation, Microsoft, and Electronic Arts. All have been incorporating their efforts and resources to figure out the benefits of using games and other interactive technologies to broaden participation in computer science.

However, because some games contain different degrees of violence and sometimes negative stereotypes against women, some researchers and educators have been incorporating other types of programming activities, such as animations and multimedia creation (Guzdial & Tew, 2006; Kelleher, 2006; Rodger, 2002) as a strategy to diversify participation in computer science and technical careers. So far, animation and multimedia creation have been producing positive impact on students' motivation and learning.

In the efforts to attract women and underrepresented minorities into technical careers such as computer science, researchers, educators, and policy makers have joined forces to implement outreach programs and different attraction strategies to encourage high school students to pursue careers in technology. These interventions have been implemented since the early 80's but little impact has been observed up until now regarding attraction and retention of more diverse students into the computer science field in North America.

Some of the intervention programs to attract high school students to learn about technology seem to be perceived as successful and seem to be creating positive experiences for some students, but they do not seem to be strongly impacting the type of students who decide to study computer science. These programs may be too short to create an effect, need more external support to be sustainable, or are motivating students who are already interested to study computer science (CS), and are thus representative of the typical CS student.

Another important strategy to address equity issues in computer science fields includes the preparation of high school teachers to create interactive software, thus allowing them to teach their students about interactive software development and encourage them to learn about technologies before entering college. Several researchers have pointed out the importance of instructing high school teachers to create different types of computer software so that they can influence their students' attitudes and perceptions about creating computer technology (Cohoon & Aspray, 2006). Summer workshops provide an opportunity to inform teachers about different technologies and applications they can use to create games and animations.

To understand how students' and teachers' interest for creating interactive software changes during and after attending two summer workshops, the researcher followed one group of students and one group of teachers who participated in a summer camp and teacher institute respectively. The summer camp represented an example of a successful outreach program designed to spark the interest of high school students in pursuing careers in science, mathematics, and technology and the teacher institute aimed to train teachers to learn the basics of computer animation by providing the tools and knowledge needed to integrate computer animation into their existing mathematics and science curricula.

Statement of the Problem

The purpose of this research was threefold: first, to determine whether an animation summer camp workshop increases high school students' interest for creating animations and games; second, to determine whether an animation teacher institute workshop increases high school teachers' interest for creating animations and games; third, to compare teachers' and students' abilities and interest in creating animations and games.

Additionally, two websites and two online communities were created for each summer workshop (summer camp and summer institute), with the purpose of providing an online presence that could support participants' learning experience and encourage the formation of an online community. Feedback was collected about the usefulness of the websites and online community.

Research Questions

Question 1: (a) How does students' interest in creating games and animations change during and after an animation summer camp? (b) How do students' previous experiences affect their interest and performance when creating interactive software? (c) What factors influenced students' interest for creating games and animations?

Question 2: (a) How does teachers' interest in creating games and animations change during and after an animation summer institute? (b) How do teachers' previous experiences affect their interest and performance when creating interactive software? (c) What factors influenced teachers' interest for creating games and animations?

Question 3: What are the differences and similarities between students' and teachers' levels of interest throughout the process of creating games and animations?

Question 4: What kind of games or animations did students and teachers create as a final project? How different was the process to create the final project for students and for teachers?

Question 5: How did teachers and students use the class website and online community?

Significance of this Study

A number of interventions have been implemented over the years to motivate girls and underrepresented minorities to learn about the use and creation of information technology, and to explore career opportunities in technology (Artemis project summer camp, 2005; girls, adventures in mathematics, engineering, and science summer camp, 2005; Seattle Smart Girls workshop, 2005; digigirlz, 2008). However, there has not been any specific analysis of changes in students' motivation to engage in the creation of graphic technology during and after the interventions based on current motivation theories and research.

Furthermore, research and interventions involving teachers creating games and animations are fairly new, as are studies on how students and teachers respond to similar teaching approaches when learning similar content. Researchers do not often have the opportunity to observe students and teachers learning at the same level, and to study how their interest for a topic develops over a short period of time.

The results from this study could help educators, game developers, computer animators, policy makers, and researchers to create models and practices that would foster students' and teachers' knowledge and interest in graphic technologies.

Summary

The present study analyzes and reports on the implementation of an animation and storytelling summer camp and a teacher institute, and explores the impact of these programs on students' and teachers' motivation to create games and animations.

Chapter 1 has presented the background of the problem, the statement of the problem, the research questions the study seek to answer; and the importance of the study for the field of computing education and gender and minority equity in technology fields.

Chapter 2 reviews the literature relevant to the study, starting with different approaches to teach game and animation creation, then two main motivation theories are introduced (self-determination and interest theory), follow by literature related to a learning model that promotes motivation. Finally, this chapter addresses the importance of online communities of learners to enhance motivation.

Chapter 3 presents the research design used for this dissertation. This includes; procedures, rationale, description of participants of the study, data collected and the process for analyzing the data.

Chapter 4 contains the quantitative results that help answer the research questions.

Chapter 5 includes qualitative analysis, samples of participants' work, some case studies about what students and teachers were able to create during the workshops, and details about how they perceived their experiences during the workshops and afterwards.

The final chapter, Chapter 6, discuses the results obtained from the summer camp and teacher institute, and compare both experiences from the perspective of motivation theories. Also discussed are limitations of the study, implications for future research, and recommendations.

Chapter 2: Literature Review

In order to get a deeper understanding of the research questions, this chapter presents theories and research regarding: strategies to teach games and animation development; motivation theories; cognitive apprenticeship model and studio thinking; and online communities of learning. The theoretical framework of this study was built upon these theories and research practices.

Animations and Games for Teaching and Learning

The main idea behind motivating students to learn throughout animation and game making is to take advantage of young adults' familiarity, engagement and fascination towards these products. Although motivating students to create technologies can be achieved in different ways, using both games and animation themes have proved to be successful in attracting the attention of girls and young students to do programming activities (Kelleher, 2006; Kafai, 1994).

Three popular approaches to motivate young adults to create graphic applications by developing games or animations are: (1) designing and programming computer games (2) using pre-made animations and 3D models to tell stories while learning to program a computer; and (3) designing and creating 3D animations to tell a story while learning the techniques and craft for creating computer animation. These three approaches are discussed in the following section.

Programming Computer Games

One of the challenges of computer science education is to create engaging ways to encourage young students to learn programming before entering college and even before starting high school. For many decades researchers have incorporated elements of computer graphics and computer game design in the programming instruction classes targeted to young students.

Seymour Papert (1980) was one of the pioneers in bringing computer science knowledge to young students, although his idea went beyond the technical aspects of programming. Instead, Papert saw programming as a way to teach strategic thinking and awareness of how we create knowledge. He designed a programming language, Logo, which allowed children to learn about problem solving, mathematics, and teaching and learning by programming a computer. Children controlled a computer cybernetic animal, the "turtle", through programming; this turtle was originally represented as a physical robotic object and later as a digital object (an icon that moves on the computer screen).

The idea was that a turtle with a pen strapped to it could be instructed to do simple things like move forward 100 spaces or turn around. From these building blocks children could build more complex shapes like squares, triangles, circles, and even draw houses or sailboats.

One of the most interesting studies on using Logo to create games was conducted by Yasmin Kafai (1994) in an inner-city public elementary school. She worked with 16-fourth grade students who were programming games in Logo to teach fractions to third graders. Students spent around 6 months working everyday for one hour on their game projects.

Kafai (1994) wanted to know what children could learn when designing games, what strategies they used to create an entertainment and educational game, and how they solved the problems they encountered on their way to achieve their goals in a long-term learning project. In the process to create these games, children learned about Logo programming, creating interaction and animation, and creating educational games.

The approach of using Logo programming in the context of creating a product was very successful according to the pretest and posttest results from Logo programming tests. When comparing the game design class scores to three different Logo classes that happened simultaneously to the game design class but did not incorporate the creation of a game, the game design class was the one that achieve better learning and motivation outcomes.

Some of the most important aspects that made this Logo game design class successful were first, the length of the program (six months) and continuity of the project; this allowed students to establish their individual styles, pace, interests, and working method. Second, students worked on creating art material and documentation for their games in their language arts class; these helped them integrate the game design activity into other areas, so students could see the process of creating a game from different perspectives. Third, students had real interaction with the end users of their games (third graders) and received realistic feedback (there were several show and tell sessions and playtest sessions during the program). Finally, students presented their games and related products to other students of the school, to parents, and to other visitors in a game fair. This approach was equally successful with girls and boys.

After Kafai's work with Logo in the early 90s, different educational programming languages have been created and used extensively to allow pre-college students create computer games. Some of the most popular language and authoring tools currently used are: Scratch (Maloney, Burd et al. 2005), Squeak (Squeakland, 2008) StarLogo TNG (Klopfer & Yoon 2005), Agensheets (Repenning 1993; Repenning & Ambach 1996; Repenning & loannidou, 2008), Alice (Hardnett , 2006), Rapunsel (Flanagan, Howe & Nissenbaum , 2005), KPL/Phrogram (Schwartz, Stagner, & Morrison, 2006), Game Maker (Gamemaker, 2008), and Microsoft XNA (Thompson, A., 2008). However, longitudinal studies focused on young students creating games have been scarce, with a few currently under way.

Programming 3D Animations

Another important approach to introduce pre-college students to programming integrates storytelling and animation. Students use programming to control the behavior of an agent and make the agent interact with different elements of the virtual environment to support the story. One of the most popular tools used to create computer animations through programming is Alice, a programming language environment developed at Carnegie Mellon that is designed for teaching programming. It uses a drag-and-drop interface and it focuses on creating virtual worlds, animated movies and simple games that can be exported to the web (Kelleher, 2006).

Although several of the tools and programming languages designed to create games discussed in the previous section support the creation of animation as well, the research team which created and maintains Alice has been re-designing Alice to support animation and storytelling more explicitly.

In an attempt to introduce girls to programming, Kelleher (2006) conducted a pilot test to study how girls engage in programming when presented with the task of creating a story using Alice.

The researcher decided to focus in storytelling for her study instead of game design because she considered that with a little time, most girls could come up with a story they would like to tell and would be able to build their stories without having to master advanced programming concepts.

There were two versions of Alice used in the study, one version was the standard "Alice 2.0" and the other was a version of Alice created by the researcher to support storytelling called "Story Alice". The purpose of this study was to compare work done using the standard Alice 2.0 version to the Story Alice version and see which version was more effective to teach programming.

The researcher randomly divided a group of eighty-eight girls into two groups; one group was assigned to use the Story Alice version and the other use the standard Alice 2.0 version. Both groups were given 2 hours and 15 minutes to go through the Alice tutorial and build an Alice world to share with everyone at the end of the workshop. Then they took an attitude survey and completed a programming quiz at the conclusion of their work, girls had 30 minutes to try the other version of Alice and decide which version of Alice they would like to take home.

The results from this study showed that girls using Story Alice spent more of their time programming instead of placing objects in the world, compared to the standard Alice 2.0 group. Girls using Story Alice were nearly three times more likely to sneak extra time to work on their programs, were more likely to chose the Story Alice version to take home and more willing to sign up for a future Story Alice class than the standard Alice 2.0 group. While, the results showed the Story Alice group was more motivated and engaged in using Alice than the standard version group, the results from the programming quiz did not show any significant differences in their scores.

Overall this was a very interesting pilot test that demonstrates the power of storytelling to engage girls with computer technology by letting them create the stories they wanted to tell. This approach was very successful with middle school girls who were very engaged in the art of telling stories through programming 3D characters. Furthermore, the game company Electronic Arts partnered with Carnegie Mellon to integrate the characters of a popular game called the Sims into Alice.

Creating 3D Computer Animations

The last approach mentioned in this section is the most direct approach to increase pre-college students' participation in computer graphic fields by teaching them the art and craft of creating professional 3D computer animations. Although this approach is the least popular of the three strategies mentioned (because of the time and resources required to implement such a course), it has great potential to diversify the game and animation industry by motivating more diverse students to pursue careers in computer graphic fields.

Computer animation classes usually have an emphasis on learning the technical aspects and art aspects of computer animation as well as learning storytelling and acting. These courses do not have a focus on programming as the previous approaches, although some programming routines are usually incorporated depending of the needs of the animation project.

Most of the documentation on teaching computer animation to learn about real animation and storytelling have been done in higher education institutions, but very little has been documented about how secondary education students perform in animation courses. Recently Smith (2006) featured an article in the *edutopia* magazine about an animation course for high school students taught in conjunction with ACME animation online communities. School teachers could teach animation in conjunction with the ACME online animation program where students learned the techniques and were guided by professors and ACME animators (group of professional animators). The ACME animators provided feedback to the students using videoconferencing. Students learned about such visual principles as design, color, drawing, and composition, but also storytelling, drama, acting, music and precision choreography. They also had the opportunity to interact with real professional animators and get meaningful feedback from them.

Among common characteristics found across different college computer animation programs is the typically multidisciplinary nature of the programs. This is accomplished by combining professors from different disciplines. For example, one class might combine computer science professors and students (Duesing & Hodgins, 2003; Cox, 1991), whereas another might combine animation and music professors and students (Orr & Nord, 2005). What it is different in these combinations is the way each discipline interacts with each other.

Storytelling is an important aspect of computer animation classes (Duesing & Hodgins, 2004; Orr & Nord, 2005; Cox, 1991), and instructors have tackled this topic in different ways. For example, Duesing and Hodgins (2004) emphasized the creation of storyboards to tell the story using drawings, photographs, and web art to visualize and explain the ideas. Orr and Nord (2005) put more emphasis on the process of understanding how to communicate an idea using animation and sound, although students also used storyboards and pitched their ideas for the story. Cox (1991) used pencil and paper techniques to instruct students about the art of creating storyboards to tell a story.

Another common aspect of these teaching animation styles is the inclusion of critique of student work. There is not too much information about how students critique each other's works; however Duesing and Hodgins (2004) mentioned that, from observations in class, computer science students were not as familiar with the critique process as art students were, so it was necessary to provide students with material about how to critique other people's work. The instructor also recommended adding critique exercises of a piece created by an animator outside the class early on in the semester. Cox (1991), on the other hand, waited until the end of the second semester to let students criticize each other's work, once they already had experience employing the basic skills.

More research is needed to evaluate how diverse pre-college students respond to a professional computer animation class to see if there are any differences in learning style, performance and motivation for these students compared to what the research community knows about college students.

The next section introduces the principal motivation theories relevant to the present research on motivating students and teachers to create interactive technologies.

Motivation Theories

Pintrich & Schunk (1996) define motivation as the process whereby goal-directed mental or physical activity is instigated and sustained. Highly motivated individuals persist in the task they are performing, even in the presence of failure. They are very focused on the task and the content they learn, and they experience enjoyment in the task.

Motivation theories offer a powerful framework to study how computer games and animations encourage diverse students and teachers to get more interested in creating interactive software. Some theories that help to understand the motivation process when creating computer games and animations are self-determination theory (Ryan & Deci, 2000) and interest theory (Mitchell, 1993; Ainley, Hidi, & Berndorff, 2002).

Self-determination Theory

Ryan and Deci (2000) proposed a theory of motivation called self - determination. This theory explains how three psychological needs for competence, autonomy, and relatedness affect self-motivation and personality. Thus, it explains variability factors in Intrinsic Motivation[1].

The human need for competence refers to the sense of feeling that one can do something well, autonomy refers to the freedom to make choices, and relatedness refers to the sense of security, caring and empathy a person experiences, the need to feel that one belongs to a group or place, and is connected with others. Self-determination refers to the process of utilizing one's will to choose how to satisfy one's needs. To be self-determining, people have to decide how to act on their environment (Pintrich & Schunk, 1996).

Some of the empirical studies that support self-determination theory showed how self-determination mediates intrinsic motivation. For example, some researchers noticed that teachers who are "autonomy supportive" (in contrast to controlling), catalyze in their students greater intrinsic motivation, curiosity, and desire for challenge

[1] Intrinsic motivation refers to the motivation or desire to do something based on the enjoyment of the behavior itself rather than relying on or requiring external reinforcement

(Ryan & Deci, 2000). Also, students taught with a more controlling approach not only lose initiative, but learn less effectively, especially when learning requires conceptual, creative processing. Similar studies with parents and their children showed that autonomy-supportive parents, relative to controlling parents, have more intrinsically motivated children (Grolnick, Deci, & Ryan, 1997). Finally, Ryan and Grolnick (1986) observed lower intrinsic motivation in students who experienced their teachers as cold and uncaring.

The importance of self-determination is also related to the introduction of the notion of "significant others" to the motivation process. Ryan and Deci (2000) stated that extrinsically motivated behaviors that are not typically interesting, sometimes emerge because the behaviors are prompted, modeled, or valued by a person's "significant others," to whom she feels (or wants to feel) attached or related. For example, Ryan, Stiller, and Lynch (1994) showed that children who had more fully internalized the regulation for positive school-related behaviors were those who felt securely connected to, and cared for by, their parents and teachers. This is a very important concept to grasp, because in order to design effective learning experiences, educators have to consider how "significant others" are related to the learners and the learning process, and how the learning environment promotes a supportive community of learners.

Nevertheless, the theory of self-determination has rarely been used to study motivation to create computer games and animation. This theory has several implications in the study of motivation and learning, and can be very useful to understand how learners relate to the environment where they learn, including the socio-cultural aspect that is embedded in creating this type of interactive software.

Self-determination theory helps researchers to think about not only the internal mechanisms happening inside the learner, but also the external aspects produced by the learning environment and the person mediating the learning process.

Another motivation theory that considers other internal and external aspects of motivation is called Interest Theory. This theory emerged from studies of intrinsic motivation and emphasizes the relationship between the learner and the content being learned.

Interest Theory

There are three types of interest identified by Krapp, Hidi, and Renninger (1992) these are: personal interest, situational interest, and topic interest. *Personal or individual interest* is considered to be an individual's predisposition to attend to certain stimuli, events, and objects. *Situational interest* refers to the likelihood that particular subject content or events will trigger a response in the moment, which may or may not "hold" over time. Thus, it refers to elicited attention in the sense of enjoyment, curiosity, and so forth, but no assumptions can be made about the level of content knowledge (Renninger, 2000). And finally, *topic interest* refers to the level of interest triggered when a specific topic is presented, and seems to have both individual and situational aspects (Ainley, Berndorff &Hidi, 2002). *Topic interest* appears when an individual's personal interest interacts with the interesting environmental features to produce the psychological state of interest in the person (Pintrich & Schunk, 1996).

The notion of interest as a way to incorporate the relationship between the learner and the object of study was also explored by Keller (Alessi &Trollip, 2001) when he proposed the ARCS model of motivation that included attention, relevance, confidence, and satisfaction as the four design considerations to create motivating instruction. However, because the interest theorists were the ones that elaborated the importance of "relevance of the content" in achievement motivation, Keller considered this element as an important factor in motivation.

Individual Interest

Several empirical studies have been conducted to explore what happens when students are exposed to material that involves topics of personal interest. For example, Renninger (2000) observed that when contexts of individual interest are inserted in expository passages, students are likely to recall more points, recall more topic sentences, write more sentences, provide more detailed information about topics read, make fewer errors in written recall, and provide additional topic-relevant information than students who read general interest passages. This is why students, whose individual interest matches school subjects, such as math, history, or geography, succeed in school.

There are also two main components associated to individual interest: stored value and stored knowledge (Renninger, 2000). These two components determine the amount of effort a person will invest, given feelings of frustration and failure, to answer questions and to resolve difficult problems.

Stored knowledge refers to a person's developing understanding of the procedures and discourse (structural) knowledge of subject content. For individual interest to emerge, a person needs to have enough knowledge to begin to organize this information in ways that bring what have been called curiosity questions. Curiosity questions help students communicate what they know and what they still do not understand; sometimes these are sophisticated questions that resemble those addressed by professionals in the content area.

Stored value refers both to a person's developing feelings of competence, and their corresponding positive and negative emotions resulting from their efforts to impose order and consistency on newly acquired understanding.

When educators and instructional designers know what types of individual interest the target learners have, they can design lessons that are sensitive to the interest area, this way they can help students ask curiosity questions and set up challenges that lead to valuing. On the other hand, if the learners have less developed individual interest, then they will be more likely to need more direct instruction designed to help create the curiosity leading to relevant questions and learning the material.

The ability to pose questions that are rooted both in what is known and in what still needs to be figured out is the basis for a person's developing knowledge about what he or she could do, or might be able to undertake, in pursing particular content. Not only does this knowledge about possible actions lead a person to challenging himself or herself to seek answers, but it also informs his or her developing sense of possible selves.

Students with less well-developed interests have a hard time engaging in activities where they do not have enough knowledge and do not have high value for the activity. Situational interest plays an important role in motivating students in this situation, because it emphasizes activities that can be done to engage students with content they are not too interested in. Fostering situational interest through activities and interesting context for the learners is important to start the motivation process when learners come to the classroom with a low interest for the subject matter.

Situational Interest

Because educators cannot easily have control over students' personal interest, Mitchell (1993) proposed a model of situational interest to help educators understand how situational interest works. His model establishes a distinction between the triggering conditions that capture the attention of the students (CATCH) and the conditions that ensure the continuation of this attention (HOLD). To catch students' interest, educators need to present the content in various ways to stimulate different people, for example, to teach about pollution, a teacher might present different activities that deal with pollution, such as showing a movie, involving students in creating a poster, and using computer games. To hold students' attention, the teachers need to empower students by helping them to find meaning or personal relevance. For example, students might teach to their communities what pollution does to the environment, or teach younger students the different sources of pollution in a big city.

Most of the research on situational interest has focused on literacy (Hidi & Harackiewicz, 2000; Hidi & Renniger, 2006; Ainley, Hillman, & Hidi, 2002; Turner, 1995; Ainley, Corrigan, & Richardson, 2005) and math education (Turner, 1998; Malone, 1981; Cordova & Lepper, 1996). These studies have examined features that make text less or more interesting, and how interesting text segments, topics, or themes influence comprehension and writing, as well as how the characteristics of the learning environment and type of instruction affect students' motivation to learn math.

From the studies on young adults' literacy it was found that there are certain themes of personal or universal significance, such as life, death, friendship, and relationships that can trigger interest, even when students have no well-developed individual interest in the domain. For example topics that touch human activity and life issues, such as body image, immediately trigger issues of concern for most young adolescents (Ainley, Hidi, & Berndorff, 2002).

The literature on situational interest and literacy describes differences between female and male students in the type of topics/activities that trigger their interest and the way they perform, when there is no well-developed interest connected to the task.

Ainley, Berndorf, and Hidi (2002) researched whether there are gender differences between interest and literary text. The researchers took four English novels and extracted several passages from the senior secondary curriculum and studied how high school students related to the readings and learned from them. The results showed significant gender differences in scores on the individual interest in literature ratings, with girls rating both literature knowledge and literature-value significantly higher than did boys. Girls reported higher topic interest and were likely to persist further with the texts than were the boys.

Another important study conducted by Hoffman (2002) combines interest and gender to investigate girls' and boys' motivation to learn physics. In this longitudinal study the researcher evaluated students' motivation towards physics from grades 5-10 in German schools.

Interests were assessed for eight topics: optics; acoustics; heat; mechanics; electricity/magnetism; electronics; structure of matter; and radioactivity/nuclear power. A questionnaire was created to combine the topics with different contexts. Some of the results from the study corroborated the findings of Ainley et al. (2002) about how certain topics initiate interest in boys but not in girls. For example girls were much more interested in learning about the mechanics of pumping blood by an artificial heart than about pumping petrol from great depths, but for boys, both were similarly interesting.

This study emphasized the importance of content and context to teach subject matters that girls found uninteresting, such as physics. Girls' responses were particularly sensitive to context. On average, girls expressed a relatively high interest in natural phenomena and phenomena that could be perceived by the senses. They placed a high value on references to mankind, social involvement, and the practical applications of theoretical concepts. The results also showed that what was interesting to girls was also interesting for boys, but not necessarily vice versa.

The findings from this study are aligned with self-determination theory, in this case, the sense of competence was the best predictor in explaining the interest in physics, and the sense of relatedness expressed by parents' support, the student-teacher's and student-student's relationship were very important factors to maintain interest in physics over time. For example, the study showed that the sense of relatedness seemed to be stronger for girls when they were in single sex educational settings, and therefore their performance was better than girls' performance in mixed gender classes.

Several of the motivation principles explained above have been integrated in a model of learning called cognitive apprenticeship (Collins, Brown, & Newman, 1989).

The Cognitive Apprenticeship Model and Studio Thinking

The cognitive apprenticeship model of learning is aimed primarily to teach the processes that experts use to handle complex tasks. The focus of this learning-through-guided-experience is on the development of cognitive and metacognitive skills, rather than on the development of physical skills and processes of traditional apprenticeships. Applying apprenticeship methods to largely cognitive skills requires the externalization of processes that are usually carried out internally. Observing the processes by which an expert listener or reader thinks and practices these skills can teach students to learn on their own more skillfully (Collins, Brown, & Newman, 1989). There are four important aspects of traditional apprenticeship: modeling, scaffolding, fading, and coaching (Collins, Brown, & Holum, 1991).

Modeling – happens when an expert is carrying out a task so that student can observe and build a conceptual model of the processes that are required to accomplish the task.
Scaffolding – includes the support the expert gives to the student in carrying out a task. This can range from doing almost the entire task for them to giving occasional hints as to what to do next.
Fading – means to slowly remove the support, giving the student more and more responsibility.

Coaching -- consists of observing students while they carry out a task and offering hints, feedback, modeling, reminders, and encouragement, working on particular weaknesses, and structuring the ways to do things.

The model of cognitive apprenticeship in formal education came from a combination of traditional apprenticeship practice in non-academic environments and the theory of situated cognition. Situated cognition theory is a theory of instruction that suggests learning is naturally tied to authentic activity, within a specific context, and culture (Brown, Collins, & Duguid, 1989). It is more difficult to learn from un-natural activities such as listening to a lecture than to learn by doing. For example, learning your first language or a foreign language by immersion is expected to be easier than learning languages from textbooks and vocabulary lists. The theory of cognitive apprenticeship provides practical steps for applying situated cognition theory (Oliver, 1999).

Characteristics of Cognitive Apprenticeship

Some of the goals of this model of learning are oriented towards the acquisition of self-monitoring and correcting skills, so the students can learn how to be more proactive and effective when it comes to getting the skills they need to learn. Another goal is to get a picture of the whole environment where the complex task they need to learn takes place. By doing this, students can also develop an interest for the skill they need to learn or to reinforce an existing interest.

In this model of instruction, students collaborate with each other and their instructor toward some shared understanding. Students have access to experts, and see other students with different skills working together. This model emphasizes the socio-cultural aspects of the learning process. In other words, learning happens when students obtain concepts and information where multiple opinions, perspectives, or beliefs must be accounted for across a group (Oliver, 1999).

Apprenticeship models are based on collaborative environments where the group works together to achieve a solution, and students enact multiple roles so that they can understand the implications of assuming a role and what it takes to accomplish the tasks and how to communicate with others. Also, as a group they confront ineffective strategies and misconceptions.

Other important features of this learning environment include tasks situated in a context that make sense to students: students are trained to look for elements that are common to the task; students learn when a skill is applicable or not; and students are able to transfer the skill to novel situations.

Besides the social aspects of the Apprenticeship model, the way tools are used in the learning context is very important. Brown, Collins, and Duguid (1989) mentioned that people who use tools actively rather than just acquire them, build an increasingly complex understanding of the world in which they use the tools, and of the tools themselves.

Cognitive apprenticeship has also been applied in wide variety of subject areas, from mathematical problem solving (Schoenfeld, 1983, 1985) to engineering (Cash et al, 1997), and has been found to be useful across a wide range of ages from elementary school students to college students.

Cognitive Apprenticeship can provide an excellent model of instruction for learning about math, science and computing through creating animation and games, but not without thoughtful consideration of several points. First, the instructors and experts involved need to get immediate feedback from the students to make sure they have properly modeled the tasks they want the students to do. This means that the instruction at the beginning of the process has to be explicit (Alessi & Trollip, 2001). Secondly, educators need to know the degree to which students are willing to become experts in a particular area of animation or games, or if students are satisfied to learn the basic skills required to become competent in the area. This knowledge might create a big difference in the effectiveness of this learning approach, as Winn (1994) states "when we can be satisfied with lower levels of achievement, situated learning is not the only, nor necessarily the best route to follow." Therefore, educators need to make sure that students want to get high level expertise on the area of games and animation they decide to work in.

Characteristics of Studio Thinking

Some of the characteristics of cognitive apprenticeship are also share by studio classroom instruction, this is a teaching framework commonly found in professional art and design classes, such as, painting, architecture, sculpting, computer animation classes, etc.

This framework was explained in a research study (Hetland, et al., 2007) analyzing the type of strengths promoted in visual arts education, i.e., what excellent visual arts teachers teach, how they teach, and what students learn. The framework explains how visual arts classrooms are structured, and the habits of minds they foster in students.

According to this framework, the structure of a typical visual arts class includes; demonstrations and lectures, individual and group projects coached by the instructor, and informal group and individual critique analysis to discuss and reflect on students' work. Similarly to the cognitive apprenticeship model coaching, scaffolding and reflecting are emphasized in the studio thinking.

The habits of mind promoted by studio thinking are development of the craft by using the appropriate tools, engagement and persistence to finish a project, envisioning what the final product will look like, conveying an idea or feeling through the art, learning to observe, reflecting, learning from mistakes, and understanding the art world.

The last section of this chapter explores the use of online communities to foster collaboration, support motivation, and increase enjoyment for the subject matter.

Online Communities of Learning

Online communities seem to be ideal tools to sustain the effort for helping students, minority students and teachers to discuss topics relate to technology with people just like them, who share the same interest, so their sense of relatedness will be reinforced by being part of a community that provides support and encouragement.

The widespread use of the Internet to foster community building has reached teenagers all over the world, particularly the teenage population in the United States;

they constitute one of the primary internet user groups in the US. Two main surveys conducted in recent years confirm this. First, the *Speak Up Report* (2005) found that students in grades 6-12 are ahead of the game with their use of technology products. They also use the newest ways to communicate, such as MySpace, Facebook, personal web pages, blogs, etc. Teenagers use technology as a natural mechanism for finding information, creating content, and shopping. Moreover, students are using technology, in particular the internet to manage their learning and social life.

According to a second recent survey conducted by the Pew Internet and American Life project (Lenhardt & Madden, 2005), 87% of teens aged 12-17 were internet users, and 51% of teenage internet users say they go online on a daily basis. In one of the most interesting findings, the number of older girls creating content on the web is increasing when compared to the number of boys. They found that older teenage girls (aged 15-17) have a strong interest in using the internet as a communication medium and for seeking and sharing information. In fact, older girls (ages 15-17) were more likely to blog than boys were; 25% of online girls in this age group kept a blog, compared with 15% of older boys who were online.

These findings encourage the use of the Internet for the purpose of building an online community of learners who can communicate with each other about their experiences in learning to create games and animations.

The communities of learners formed around professional activity or work practice are referred as communities of practice (Wenger, McDermott & Snyder, 2002) formed by "groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis." Some of the features of community of practice can be intentionally included when creating online communities to learn about animation and games. Communities of practice also help members build a sense of professional identity by sharing knowledge and experiences.

In the following chapter, some of these theories are presented in the context of the summer workshops. The research methodology is explained as well as the process to collect and analyze the data.

Chapter 3: Research Method

The purpose of the research was to analyze how students' and teachers' interest in creating games and animation technology changes over time, with respect to how previous experiences affected their learning experience, and how the learning environment and online community affected their interest for creating interactive software. The two workshops included a summer camp for students and a summer institute for teachers; the results from both interventions were analyzed and compared.

A mixed method approach was used to collect and analyze both quantitative and qualitative data from the summer workshops. Within-group measures were analyzed to study the learning experience of the students and, subsequently, the experiences of teachers followed by a between-group analysis to compare the experiences of high school students and high school teachers. Between-group analysis was based on differences and similarities in the way each group engaged with the subject matter and the type of applications they created as final projects.

The Summer Workshops

The summer workshops were part of a three-year educational outreach project designed to spark the interest of high school and college students in pursuing careers in science, mathematics, and technology. The project was especially directed to young women and other groups who have traditionally been underrepresented in these fields. The two main project activities were (1) a two-week summer camp for high school students and (2) a two-week summer institute for high school teachers.

Students and teachers attending the summer workshops were presented with the basic tools used to create an animation or interactive game from scratch, including a programming language called Visual Basic, and graphic editors such as Microsoft Paint and Adobe Photoshop. Participants received an overall perspective of what it takes to create games and animations, as they learned basic graphic programming and image editing skills.

The summer camp included high school students mainly from the Seattle area, but the summer institute included math, science, and social studies high school teachers from the Pacific Northwest. Content and instructors were the same for both camp and institute, the main difference being the type of final project required for each group.

Students were required to create an animation or game as final project whereas teachers were expected to create an educational software or educational game.

Description of the Summer Workshops

The summer camp took place the first two weeks of July 2007 and the teacher institute took place the last two weeks of July 2007. The settings for both were in educational institutions in the Seattle area. Flyers were sent to schools in the area, and a website advertised the workshops. Twenty-four students and twenty-three teachers were selected on the basis of their online application, which included personal statements about why they wanted to be participants.

The summer camp was free of charge, and minorities and girls were highly encouraged to participate according to the camp website. The project provided all the materials the students used including computers, and all students received notebooks, thumb drives, handouts, and five college credits for completing the workshop. The project also provided food and gifts such as software, t-shirts and laptop bags.

The teacher institute was not only free of charges, but also provided a compensation of $700 for each teacher who completed it. The summer workshop encouraged teachers who wanted to teach math and science through games and animation to apply. Teachers' meals and materials were provided, plus they received five graduate education credits or equivalent clock hours, as well as diverse gifts.

The summer workshops consisted of one programming instructor, one art teacher, and five Teaching Assistants (TAs), who were college students. The researcher was one of the TAs. At any given time, there were at least one instructor and four TAs present. Two computer classrooms were available for the students, and one computer classroom was available for the teachers, because the summer camp and teacher institute took place in different physical locations in the Seattle area.

The summer camp's days started at 9:30am and ended at 3:30pm. During the first week participants were introduced to basic computer graphics programming using Visual Basic and how to create background images and characters in Photoshop, to create a game-like environment. During the second week, participants learned more advanced computer graphic programming techniques, such as adding interactions and sounds in their programs, and completed their final project. Students worked in their final project for a few hours during the last couple days before the final presentation. At the end of the summer camp, parents and families were invited to watch as each student group or individual student presented their final creations to the audience.

The summer institute for teachers followed the same daily schedule from 9:30am thru 3:30pm. Most of the instruction on learning how to do basic graphics programming was presented during the first week. The second week was mostly devoted to creating the final project, which consisted of an educational software or game that taught some subject matter content, and a lesson plan to go with it. The teachers' projects and work were graded, and the institute ended with them presenting their final project to the class, explaining what they did and showcasing their software.

Two websites were created, one for the summer camp and one for the summer institute. They provided tutorials about how to do graphic programming, access to free images and sound, "galleries" to show participants' work and final project, and a link to an online Google Groups where they could interact with their fellow camp or summer institute colleagues.

Summer Workshop Participants

The criteria to select participants for the summer camp and teacher institute were determined by the workshop director and advisory committee. The researcher was not involved in developing the criteria, or selecting the participants. She was able to view the online application form participants needed to fill out and got hints about the selection process from the director of the summer workshops.

Summer Camp Recruitment Criteria

To be eligible, students had to be in high school or entering high school in the fall and submit an online application that asked for personal information such as grade level, gender, age, name of math or science teacher, race/ethnicity (optional), college plans (optional), and what they wanted to study in the future (optional). Additionally, students had to write a short statement indicating the reasons they were interested in attending the arts and animation camp, and how their personality, achievements or potential would enable them to benefit from the camp.

According to the director of the program, the criteria used to select participants included students who were not usually engaged with technology, mathematics and science, but were interested in creating computer animation, and had little, if any, experience creating computer games. Students who wrote in their application essay that they wanted to create games or indicated they were already familiar with technology, were excluded. Because a primary goal of the project was to encourage girls to consider careers in computer science and related fields, the percentage of students in the camp was about 60% female and 40% male students.

The summer camp website application explained that student participants would be selected by a committee on the basis of personality, evidence of achievement, and their desire to participate in the camp indicated in the short statement of interest required with the application. Girls and other students from underrepresented communities in science were strongly encouraged to apply.

The total number of students accepted for the camp was 24 (15 female and 9 male students), which met the desired female-male ratio. However, of all the students, only two were African-American, four were Asian and there were no Latino students.

Summer Camp Research Participants

Fifteen of the 24 students volunteered to be part of this research study, including 10 female and 5 male students from 12 different high schools in the Seattle area. To participate in the study, students could not have created a computer game before, were willing to volunteer, and had to obtain their parents' written consent to let them participate in the study. Three students did not complete some of the key surveys or were not able to provide the programs they created during the camp. As a result the final sample consisted of 12 students, 8 female and 4 male students. The students' age in the final sample ranged from 13 to 16 years ($M = 14.92$, $SD = 0.95$). None of the study participants was African-American.

Summer Institute Recruitment Criteria

For the teachers to be eligible to participate in the summer institute they had to be teaching 9-12th grade math, science, or technology; college teachers were also able to participate. No previous computer programming experience was required or expected. Teachers from the same institution were encouraged to apply together.

Teachers were also required to submit an online application that asked for personal information, such as, name, address, grade level taught, major subjects taught, school, school district, indicate if they were applying with another teacher from their school, and indicate their level of expertise with computers from the following options: a) Email and web only; b) Reasonably proficient with applications and file management; c) Very experienced, including some programming experience.

Additionally, two essays needed to be submitted online: a short statement (maximum 500 words) describing why they were interested in attending the Teacher Institute and what they were hoping to gain from the experience; and a short statement (maximum 200 words) describing how they could use the animation module(s) they created in the summer institute in their classroom. If they were applying as a team, they needed to include some comments on how they would work together.

According to a member of the program committee, teacher participants were expected to have some familiarity with using computers. However, remarkable differences in participants' familiarity and knowledge about using computers were observed during the research and project.

There was no goal related to the gender, race, or ethnicity of the teachers. The total number of high school teachers accepted for the summer institute was 23 (13 females and 10 males).

Summer Institute Research Participants

All the teachers enrolled in the summer institute volunteered to be part of the research. The sample included 13 female teachers and 10 male teachers.

The main condition for participating in the study was having limited programming experience, especially not having any game programming experience. Two participants were experienced programmers: one had a PhD in computer science; the other had a bachelor's degree in computer science, plus 10 years of software development experience. Because they had very limited experience creating games and animations, some of their data were still included in the study.

Twelve teachers had applied with coworkers, resulting in six pairs of teachers who were expected to work together. Eleven teachers came from different schools, and they could either work alone or in pairs to create the final project. Teachers' grade level taught varied from 7th thru 12th grade, and also included one college professor. The subjects they taught in school ranged from computer programming, computer fluency, biology, mathematics, physics, social studies, science, and chemistry.

Instruments

Five surveys were created for each workshop. The summer camp and summer institute surveys were very similar and designed to gather information related to the participants' level of interest with respect to creating interactive software. The surveys were available online, and participants were given a time frame for completing each one. Only the researcher had access to the survey results and anonymity of participants was maintained.

Students' Instruments

The study used a questionnaire to gather basic demographic information and four survey instruments to gather data related to student attitudes and evaluations about computer technologies and the workshop (Appendix A). The questionnaire asked for students' name, nickname (created by combining their favorite animal with their month and day of birth) to keep confidentiality, age, the name of the school, and their grade level in the fall of 2007. The four survey instruments are described below.

Students' interests in graphic technologies (Survey 1)

This questionnaire (see Appendix A) was developed specifically for this study. The design was guided by a variety of instruments used by researchers studying students' interests in information technology and computer games (Cohoon & Aspray, 2006; Kafai, 1994). The questionnaire included 25 questions related to level of interest in: animation and digital games; computers, mathematics, drawing, and the summer camp itself.

Programming self-concept questionnaire (Survey 2)

Nine items were used to assess attitudes towards programming, such as their perception of the level of effort required to do programming, level of interest, and degree of self-efficacy. These were adapted from a computer science motivational scales survey by Pajares, Cheong, and Oberman (2004). (See Appendix A).

Feedback and interest changes survey (Survey 3)

This survey contained 12 questions and asked participants to provide feedback related to the summer camp. It was designed to evaluate their current interest in creating games and animations after their workshop experiences; what they enjoyed about the camp; what they did outside the camp to complete their final project; and their evaluation of the website developed to support the summer camp activities (See Appendix A).

Post-summer camp survey (Survey 4)

The fourth survey included eleven questions for the students to assess their current level of interest in creating animations and games, the probability that they will continue to pursue the animation and game work initiated in the camp, their school's ability to support continued learning about technology, and actions they had taken to further their knowledge (See Appendix A).

Teachers' Instruments

A questionnaire was used to gather basic demographic information from the teachers, such as, teachers' name, nickname (created by combining their favorite animal with their month and day of birth), grade level they taught, subjects they taught, and school where they taught. Also four survey instruments were designed for this part of the study (Appendix B). The four survey instruments are described below.

Teachers' Interests in graphic technologies (Survey 1)

This survey was similar to the student's interest in graphic technology survey, with minor adjustment to address teachers' experiences. For example, instead of asking about their self-efficacy related to mathematics, they were asked about their self-efficacy related to teaching mathematics, science and technology. This questionnaire contained a total of 25 questions also related to their level of interest in animation, digital games, and computers, mathematics, and drawing; their expectations and interest in the summer institute; and their access to computers at home and how they used them (See Appendix B).

Programming self-concept questionnaire (Survey 2)

The same questionnaire described in the students' instruments section was used for the teachers (See Appendix B).

Feedback and interest changes survey (Survey 3)

Essentially, the same twenty-two-question survey for the students was used for the teachers. Teachers were asked to evaluate their current level of interest in creating games and animations; which summer institute experiences they enjoyed; what activities they did outside the summer institute to complete their final project; and to provide feedback about the website developed to support the summer institute activities and the online community (See Appendix B).

Post-summer camp survey (Survey 4)

This survey included eleven questions related to their current level of interest in creating animations and games; the probability that they will continue their animation and games work initiated in the camp; their ability to integrate what they learned in the summer institute into their classes; feedback received from their students when using the games and animation teachers created; use of the summer institute website and online community; and actions taken to further knowledge (See Appendix B).

Procedures and Data Collection

All the high school students received the same instruction during the summer camp. Although, they were encouraged to work with a partner for their final project, it was not mandatory. The curriculum for the summer institute was very similar to the one for the summer camp. The only difference was the type of final project teachers were asked to do at the end; all the teachers received the same type of instruction as well and the same type of encouragement to work in pairs for their final project.

The researcher acted as a participant observer during the workshops. As a participant, she served as a teaching assistant during the summer camp and teacher institute. She also created the class website as an additional resource for learning the process of creating games and animations. The class website included video tutorials that explained the graphic programming task they were doing in class using Visual Basic, screenshots of participants' programming tasks, and videos and source code for the final project. In addition to creating and updating the summer workshops' websites, the researcher also created and maintained the online community for students and teachers.

In her role of researcher and observer, the researcher reminded participants to take the online surveys, took field notes of the instruction method and participants reaction to the instruction and programming tasks, and interviewed the instructors to get their perspective about their experiences in the workshops.

The procedure to collect the data was the same for the summer camp and teacher institute. The order in which each set of data was collected is described below.

Surveys

During the second day of the workshops, the invitation letters to participate in the study and consent forms were given to participants. Students were required to have their parents sign the consent forms if they decided to participate in the study. Once the consent forms were collected, a note with the link to the first survey was provided to the participants. They were asked to fill out the background information and first survey in order to capture their initial interest and previous experiences with technology, mathematics and art.

Survey 2, which addressed their perceptions about how difficult it was for them to do the programming tasks, and how competently they felt they performed them, was available online the second week of the workshop and institute.

Survey 3 collected feedback from participants' experiences in the workshops; this was available online the last day. Participants either filled it out during the last day of the workshop or later from home. Three email reminders were sent to the participants.

Survey 4 was not available online until October 2007. Its purpose was to determine if participants' interest in creating interactive software had changed, and if they had been able to engage in further learning experiences that had expanded their knowledge regarding interactive software. The researcher also wanted to determine if teachers had been able to integrate what they had learned in the summer institute into their classroom practice.

Field Notes

The researcher took Field notes every day during the summer camp and teacher institute. In addition to recording the activities and responses, the researcher was specifically looking for the following:

- Instances of apprenticeship instruction,
- Questions asked by participants,
- Difficulties participants experienced trying to accomplish the programming tasks,
- Participants' engagement and persistence with the tasks,
- Collaboration among participants, and
- Type of help provided by the instructors and TAs to the participants to complete their assignments

Programming Assignments

Three sets of Visual Basic programming tasks done in the classroom were collected from each participant for research purposes: (1) source code, (2) images and (3) sounds.

The first assignment involved creating a scene that included a house, skyline, clouds, rain falling from a cloud, the sun, a lake, and a person or animal from a previous art activity, and was collected the second day of class. Participants had to use the basic shapes provided in Visual Basic to create the elements of the scene, such as, lines, rectangles, ellipses and polygons, in addition to using textures. To accomplish this task effectively, they needed to use their knowledge about the computer's coordinate system to place objects properly, as well as understand how the sequence of the code they wrote would affect the way the objects would appear in the scene. This assignment was the same for the students and the teachers attending the workshops (Figure 1).

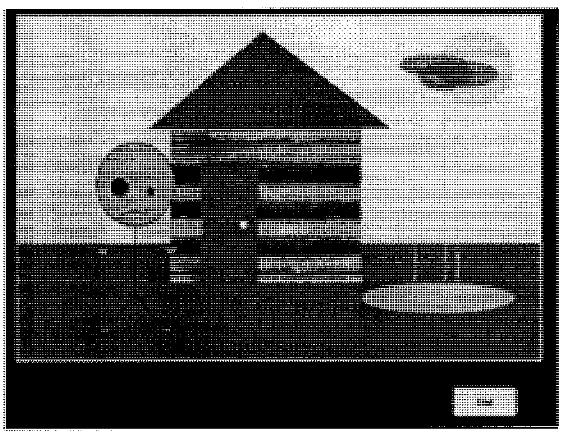


Figure 1: Screenshot of Programming Assignment 1.

The second programming task done in the classroom was collected at the beginning of the second week of class. This task was the core of student's final project, but did not serve the same purpose for the teachers. Furthermore, students and teachers' second assignment were different.

The requirement for this programming task was to add a character to a background image previously created in the Photoshop class. In addition, participants had to make the character appear to walk, and add background music and sound affect. Finally, a storyline in the form of text had to be created. To be successful in this task, students needed to follow the instructions about how to make the background scroll. This involved preparing the background image in the photo editor to create continuous, smooth background scrolling without any sudden discontinuity in the image. They also had to learn how to program keyboard events, add music through the windows media player tool, and effectively use timers to incorporate sound effects and changes in the story line (Figure 2).

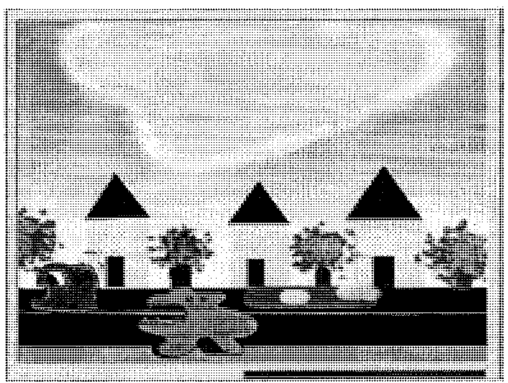


Figure 2: Screenshot of a Student's Programming Assignment 2.

The teachers' second assignment was simpler than students' second assignment. They had only to integrate a background image with a character they had either created or found on the Internet. They were required to make their character respond to users' keyboard input, and provide some background music and sound effects.

To be successful in this task, teachers needed to program keyboard events, add music through the windows media player tool, and effectively use timers to incorporate sound effects and changes in the story line (Figure 3).

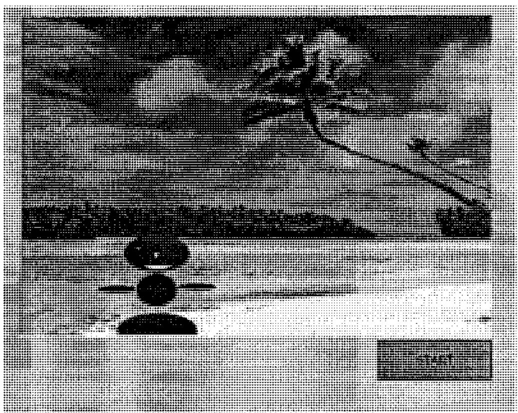


Figure 3: Screenshot of a Teacher's programming Assignment 2.

The last programming assignment collected from both groups was the final project. The final project requirements were also different for students and teachers. Students were required to create an animation or game, preferably that was a continuation of what they did for their programming task 2 (Figure 4).

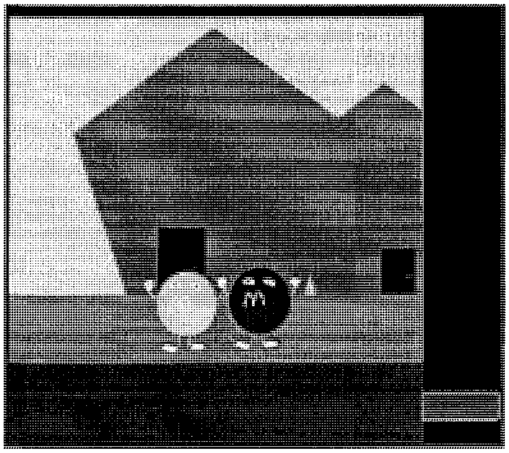


Figure 4: Screenshot of a Student's Final Project.

The teachers' final assignment had a specific set of requirements that were given to them in a handout. They needed to create an educational software or game that they could use to teach a lesson or principle about their subject matter at own school. Whichever they chose, it had to incorporate sound, an animated character, and support the user's keyboard or mouse input. Additionally, teachers were required to submit a written lesson plan with learning objectives, expected outcomes, their plan for integrating it into their curriculum with possible extensions, and potential barriers to using their game or animation in their classroom (Figure 5).

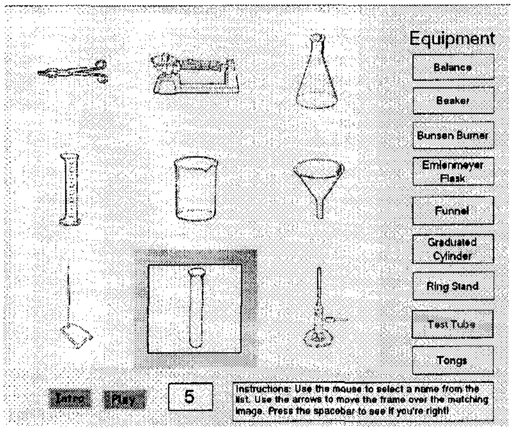


Figure 5: Screenshot of a Teacher's Final Project.

Instructors' Interviews

Workshop instructors and TAs were asked six questions during short, individual interviews at the end of each summer workshop. One instructor and one TA did not complete the interview, resulting in a final set of data from seven interviews. In each interview, the researcher asked the following questions:

1. What went well or wrong during the workshops?

2. What challenges did they experience?

3. What kinds of help did they provide to the participants?

4. How competently did they think the group of students or teachers performed in the programming tasks?

5. Did they use any extra materials to prepare for the class?

6. Did they do anything in particular to keep participants motivated?

Participation in the Online Community

The researcher monitored the online community to collect new material posted by the participants, because that was where they could post new art work, programming code, questions and suggestions. However, some participants were more comfortable with sending their work by email, so in some cases their new programming samples were collected this way.

Data analysis

Overall the researcher was looking for evidence of participants' interest for designing and developing interactive software, and how both their previous experiences and the learning environment of the workshops may have mediated changes in their interest towards creating animations or games.

For the purpose of this research, interest was operationalized as a combination of positive attitudes towards animations or games, some knowledge about how games or animations are created, persistence in the task and desire to learn more about creating games or animations.

Factors predicted to create, maintain or increase students' interest include: performance on programming tasks, positive attitudes towards their learning experiences, high sense of self-efficacy towards programming, and a sense of connection with the people participating in the learning environment.

Previous knowledge relevant to the creation of animation or games, such as being skilled at mathematics, creating digital art, and general computer use, also might impact performance and attitudes towards the workshop tasks, because graphic programming involves a combination of mathematic concepts, digital art skills, and computing skills.

To answer the research questions, repeated measure analysis of variances were conducted on students and teachers' responses to their initial interest to create animations and their interest to create games at three different points on time, at the very beginning of the summer classes, at the end of the summer classes, and two months after summer classes ended.

Pearson correlation studies were conducted among participants' rating of sense of competence for using computers, sense of competence for doing/teaching math, subjective score for the first and second programming task given by the researcher, ratings of self-efficacy for programming, perceived effort to do programming, interest in doing programming, and positive attitudes towards doing programming.

Qualitative analyses were carried out using six case studies of participants who exhibited different ranges of skills and motivations to create games and animations. Participants' previous knowledge, positive attitudes, sense of relatedness towards the camp, performance on three programming tasks, collaboration with their project partner (if any), persistence and engagement to continue working on the project were all categorized and analyzed based on surveys, researcher's field notes, programming software collected, and instructors' interviews.

Comparisons between-group (students and teachers) provided answers about differences and similarities on participants' interest for creating games and animations, participants' attitudes, self-efficacy, performance, and previous experiences.

Finally, a qualitative comparison of how students and teachers responded to the method of instruction was conducted based on the researcher's field observations and comments from the instructors and TAs' during the interviews.

A qualitative approach was used to evaluate participants' final project. A comparison of the types of software they created was made based on the sophistication of programming and number of lines of code required.

The process participants used to create the final project was analyzed by looking at participants' progress among the three programming tasks collected, field notes about their work flow, participants' feedback provided in the survey, and instructors and TAs' comments from the interview.

Quantitative and qualitative analysis were used to understand the usefulness of the class website and online community. Participants' feedback from the surveys and instances of participation in the online community were the main data for the analysis.

In the next chapter, the results obtained from the quantitative and qualitative analysis are presented and analyzed.

Chapter 4: Results

This chapter presents the results obtained from the summer camp and summer institute. Quantitative analyses were conducted and explained by framing the results around the research question under investigation. A qualitative analysis is presented in Chapter 5.

All the data analyzed in this section came from participants' surveys, and three programming assignments each person created during the workshops. Correlation analysis, comparison of two means using t-tests, and analysis of variance (ANOVA) were the main statistical analyses utilized to interpret the data. SPSS version 13 was used to perform all the computation and to derive graphics presenting the data.

Results from Student Participants

Question 1: (a) How does students' interest in creating games and animations change during and after an animation summer camp? (b) How do students' previous experiences affect their interest and performance when creating interactive software? (c) What factors influenced students' interest for creating games and animations?

(a) How does students' interest in creating games and animations change during and after an animation summer camp?

To answer the first part of question 1, the researcher asked students at three different points in time (beginning of the summer camp, end of the camp, two months after the camp ended) how much they were currently interested in creating games, and how their current interest for creating animation was. Answers were collected through online surveys and the results were ratings from 1 to 5, where 1 indicated they were not interested at all in creating games or animations, and 5 meant they were very interested.

Table 1: How strong is your interest in creating animations? (Students)

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Pre-Camp	12	3.7500	.75378	.21760	3.2711	4.2289	3.00	5.00
Post-Camp	12	3.7500	.75378	.21760	3.2711	4.2289	3.00	5.00
Follow Up	11	3.7273	.78625	.23706	3.1991	4.2555	3.00	5.00
Total	35	3.7429	.74134	.12531	3.4882	3.9975	3.00	5.00

Table 2: How strong is your interest in creating computer games? (Students)

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Pre-Camp	12	3.4167	1.24011	.35799	2.6287	4.2046	1.00	5.00
Post-Camp	12	3.0833	1.31137	.37856	2.2501	3.9165	1.00	5.00
Follow Up	11	3.2727	1.19087	.35906	2.4727	4.0728	1.00	5.00
Total	35	3.2571	1.22097	.20638	2.8377	3.6766	1.00	5.00

Before proceeding to analyze the data using analysis of variance (ANOVA), the assumptions to perform that test were checked, and the results are displayed on Table 3, 4, 5 and 6. First the researcher observed if the data was normally distributed and then if the homogeneity of variance was achieved.

Table 3: Tests of Normality for Interest in Animation (Students)

	Kolmogorov-Smirnov			Shapiro-Wilk		
	Statistic	df	Sig	Statistic	Df	Sig
Pre-Camp	.257	12	.028	.807	12	.011
Post-Camp	.257	12	.028	.807	12	.011
Follow Up	.277	11	.018	.799	11	.009

Table 4: Tests of Normality for Interest in Games (Students)

	Kolmogorov-Smirnov			Shapiro-Wilk		
	Statistic	df	Sig	Statistic	Df	Sig
Pre-Camp	.215	12	.132	.903	12	.172
Post-Camp	.225	12	.096	.910	12	.211
Follow Up	.228	11	.116	.916	11	.285

Table 5: Test of Homogeneity of Variance for Interest in Animation (Students)

Levene statistic	df1	Df2	Sig.
.035	2	32	.966

Table 6: Test of Homogeneity of Variance for Interest in Games (Students)

Levene Statistic	df1	Df2	Sig.
.043	2	32	.958

The sample distribution for the results obtained from interest in creating animation were not normally distributed ($\alpha = 0.05$), see Table 3, and the ones for interest in games were normally distributed ($\alpha = 0.05$), see Table 4.

The assumption for homogeneity of variance was accomplished for the interest in animation sample and interest in games sample ($\alpha = 0.05$), see Tables 5 & 6.

Despite the fact that some of the data were not normally distributed, ANOVA was used for both data sets (interest in animation and interest in games), according to Steckler & Oleson (2005) ANOVA is robust enough over moderate violations of the distribution assumption.

One-way ANOVA was conducted to examine whether students' interest for creating animations was stronger after the summer camp ended compared to the beginning of the summer camp, and whether students' interest for creating animations had faded, increased or decreased two months after the camp ended. The results obtained are shown in Table 7.

Table 7: ANOVA for Interest in Creating Animations (Students)

	Sum of Squares	Df	Mean Square	F	Sig.
Between Groups	.004	2	.002	.003	.997
Within Groups	18.682	32	.584		
Total	18.686	34			

A separate one-way ANOVA was also conducted to examine students' interest in creating computer games. The results are shown on Table 8.

Table 8: ANOVA for Interest in Creating Computer Games (Students)

	Sum of Squares	Df	Mean Square	F	Sig.
Between Groups	.671	2	.335	.215	.808
Within Groups	50.015	32	1.563		
Total	50.686	34			

Form the results obtained from ANOVA there was no significant effect on interest in creating animations after the summer camp ended or two months after the summer camp ended, $F\ (2,32) = .003,\ \rho > .05$.

Also there was no significant effect on interest in creating animation over time, $F\ (2,32) = .215,\ \rho > .05$.

Students were asked after the summer camp ended if they were planning working on their animations in the future. Twelve students out of twelve responded the survey; the results are shown in Figure 6.

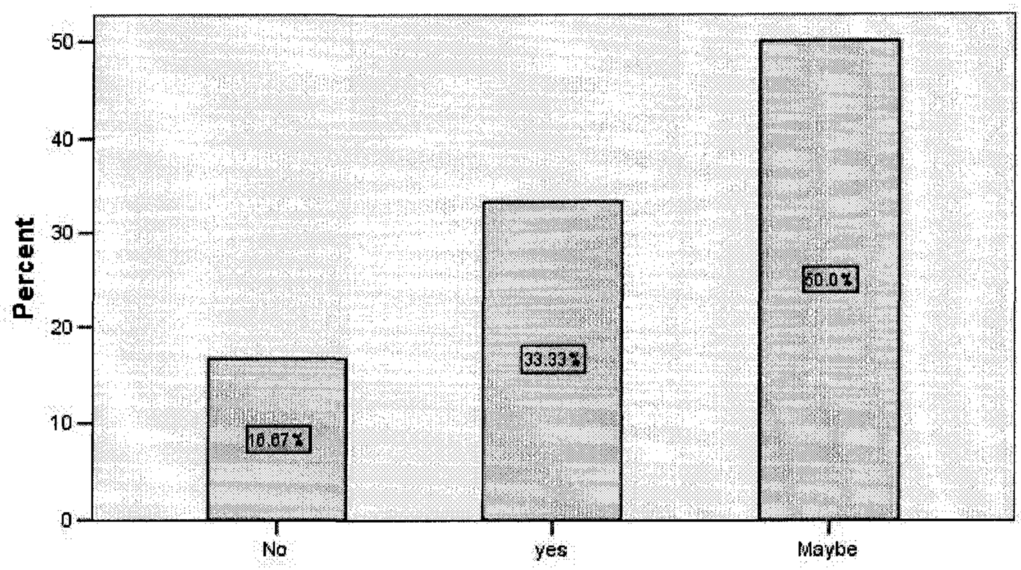


Figure 6: Are you planning on continuing working on your animation project at home or school?

The results from the survey showed that 16.67 % did not plan working on their projects anymore, 33.33% planned continuing working on their project, and 50% thought they mightwork on their project.

Two months after the summer camp ended students were asked in a follow-up survey if they indeed worked in their animation and game projects after the camp ended and the responses are shown on Figure 7. Eleven students out of twelve responded the survey.

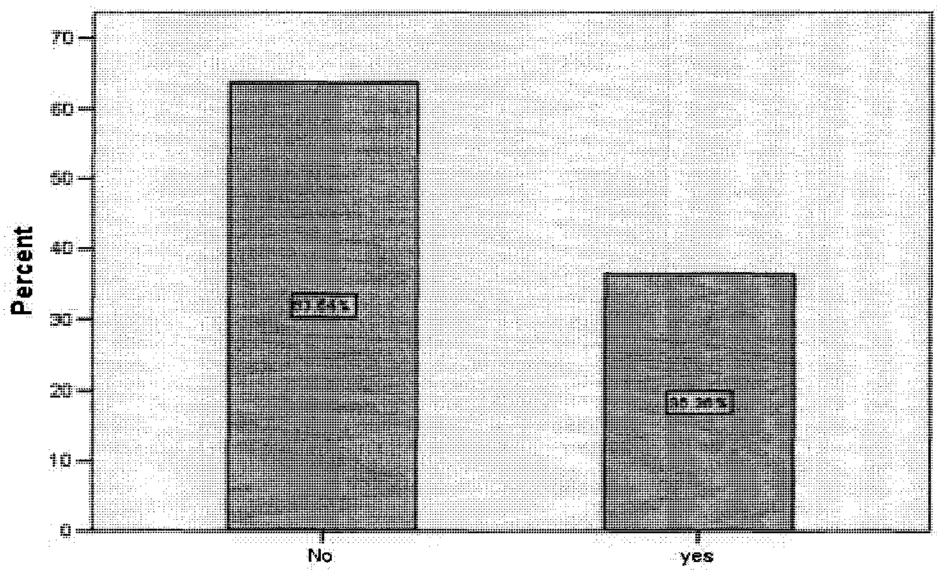


Figure 7: Did you keep working in the final project you created in the summer camp 2007?

The results showed 63.64% students did not continue working in their animations while 36.36 % students did continue working on their projects.

Finally, students were asked in the same follow up survey if they had created any new animation or game after the camp ended. Figure 8 show the percentage of students who did create a new game or animation versus the percentage of students who did not.

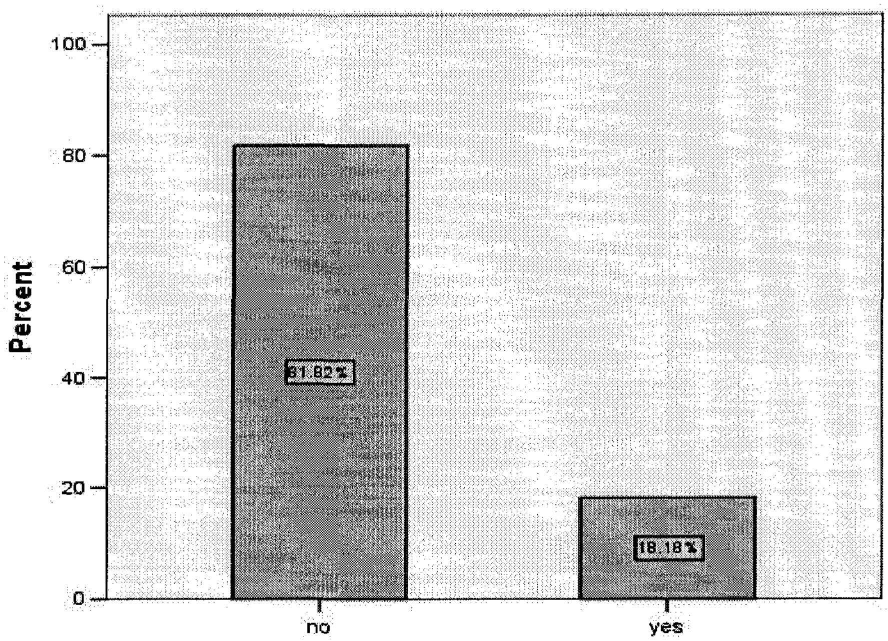


Figure 8: Have you created any new game or animation after the camp ended?

A majority of 81.82% of the students did not create any new animation or game, and 18.18% created one.

(b) How do students' previous experiences affect their interest in programming and task performance when creating interactive software?

To answer this question, results obtained from a programming attitudes survey, pre-summer camp survey, and subjective ratings for programming task 1 and programming task 2 were analyzed using Pearson correlation (see Table 9).

The programming survey, given a few days before the camp ended, asked students to rate on a scale of 1 to 5 how comfortable they were at programming, how interesting and challenging they thought programming was, how good they were at programming, how excited they were about programming, how much trouble they had with the programming tasks, how easy it was learning to program, and how well they did in the programming tasks.

The subjective scores for the programming task 1 ("create an scene using geometric shapes and textures), were based on two criteria: a) how well they understood the computer coordinate system to properly place objects on the screen, and b) how well they understood the commands to display objects on the screen using programming.

Programming task 2, which consisted of creating a background scrolling effect and adding a character in the scene, was graded based on several criteria that account for understanding how to create interactive programs. The score was based on: a) ability to integrate a character or sprite and animate it, b) ability to incorporate a scrolling background, c) successful integration of music and sound effect, d) incorporation of storyline or instruction/feedback screens in the game or animation, e) ability to implement different keyboard input to allow users to interact with the character in the game/animation, and f) good use of timers and correct implementation of them.

From the pre- summer camp survey, students were asked about the depth of their knowledge of computer use, and how good they were at doing mathematics.

Table 9: Pearson correlation for students' programming attitudes, performance on task, knowledge about computers and perceived self-efficacy towards math (significant correlations are highlighted)

Questions -->	1	2	3	4	5	6	7	8	9	10	11	12
Students (N=12)												
1=comfortable	--	.24	-.74	.67	-.07	-.56	.70	.62	.48	.41	.48	.29
2=interesting		--	-.57	-.13	.60	-.11	.00	.00	.00	.57	.57	.09
3=challenging			--	-.53	-.29	.40	-.76	-.64	-.29	-.56	-.66	-.48
4=good at				--	.16	-.51	.70	.72	.66	.36	.34	.72
5=excited					--	-.07	-.16	-.06	.12	.40	.44	.41
6= trouble						--	-.41	-.68	-.66	-.24	.16	-.33
7=easy to							--	.78	.32	.35	.30	.54
8=good in tasks								--	.59	.31	.18	.63
9=Task 1									--	.57	.25	.68
10=Task 2										--	.67	.50
11=Computers											--	.41
12= Math												--

There was a negative relationship between how comfortable students feel about programming and how challenging they thought programming was, $r = -.74$, $\rho < .01$.

Conversely, there was a positive correlation between how comfortable they were doing programming and how good at programming they were, $r = .67$, $\rho < .05$, as well as, how easy it was to do programming and how comfortable they were $r = .70$, $\rho < .05$. Students also perceived themselves as doing well in the programming task when they were comfortable doing programming, $r = .62$, $\rho < .05$.

A positive correlation was found between how interesting programming was and how excited students were about programming, $r = .60$, $\rho < .05$.

Amount of challenge to do programming was negatively correlated with how easy it was to do programming, $r = -.76$, $\rho < .01$, and negatively correlate with being good on the programming tasks, $r = -.64$, $\rho < .05$. Negative correlation was also found between how challenging programming was and how much computer experience students had, $r = -.66$, $\rho < .05$.

Feelings of competence about how good students were at programming proved to be positively correlated with how easy they thought programming was, $r = .70$, $\rho < .05$, positively correlated with how good they were on the programming tasks, $r = .72$, $\rho < .05$, and positively correlated with how good they performed on programming task 1, $r = .66$, $\rho < .05$. Finally being good at programming was positively correlated with being good at math, $r = .72$, $\rho < .01$.

Having trouble with the programming tasks was negatively correlated with how good they felt they were doing in the programming tasks, r=-.68, ρ <.05, and negatively correlated with performance on task 1, r= -.66, ρ <.05.

Feelings of doing well in the programming tasks were positively correlated with being good at math, r= .68, ρ <.05.

Finally performance on programming task 1 was positively correlated with doing well at math, r=.68, ρ <.05, and performance in task 2 was positively correlated with experience using computers, r=.67.

(c) What factors influenced students' interest for creating games and animations?
This question is answered in chapter 5, when a qualitative analysis is presented and explained.

Results from Teacher Participants

Answers for question number 2 related to responses from high school teachers were analyzed using the same approach as that used for the students.

Research Question 2: (a) How does teachers' interest in creating games and animations change during and after an animation summer institute? (b) How do teachers' previous experiences affect their interest and performance when creating interactive software? (c) What factors influenced teachers' interest for creating games and animations?

(a) How does teachers' interest in creating games and animations change during and after an animation summer institute?

Teacher participants were also asked about their current interest in creating games and animations at three different times. Their responses were collected through three different online surveys at the beginning of the summer institute, at the end, and two months after the summer institute ended. The results were expressed as ratings from 1 to 5, where 1 meant they were not interested at all in creating games or animations, and 5 meant they were very interested. See Table 10 and 11.

Table 10: How strong is your interest in creating animations? (Teachers)

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Pre-Inst	22	4.2273	.86914	.18530	3.8419	4.6126	2.00	5.00
Post-Inst	19	4.4211	.90159	.20684	3.9865	4.8556	2.00	5.00
Follow Up	16	3.5000	1.15470	.28868	2.8847	4.1153	2.00	5.00
Total	57	4.0877	1.02261	.13545	3.8164	4.3591	2.00	5.00

Table 11: How strong is your interest in creating computer games? (Teachers)

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Pre-Inst	22	3.9091	1.06499	.22706	3.4369	4.3813	2.00	5.00
Post-Inst	19	4.2632	.93346	.21415	3.8132	4.7131	2.00	5.00
Follow Up	16	3.6250	1.14746	.28687	3.0136	4.2364	2.00	5.00
Total	57	3.9474	1.05933	.14031	3.6663	4.2284	2.00	5.00

The data was also checked to verify that ANOVA assumptions about normal distribution were not violated. First the researcher observed if the data was normally distributed and then if the homogeneity of variance was achieved. See Tables 12, 13, 14, and 15.

Table 12: Test of Normality for Interest in Animation (Teachers)

	Kolmogorov-Smirnov(a)			Shapiro-Wilk		
	Statistic	Df	Sig.	Statistic	df	Sig.
Pre-Interest Animation	.288	15	.002	.783	15	.002
Post-Interest Animation	.353	15	.000	.728	15	.001
FollowUp-Interest Animation	.173	15	.200(*)	.876	15	.042

* This is a lower bound of the true significance.

Table 13: Test of Normality for Interest in Games (Teachers)

	Kolmogorov-Smirnov(a)			Shapiro-Wilk		
	Statistic	Df	Sig.	Statistic	df	Sig.
Pre-Interes Games	.311	15	.000	.737	15	.001
Post-Interest Games	.318	15	.000	.778	15	.002
Follow-Interest Games	.214	15	.062	.859	15	.023

Table 14: Test of Homogeneity of Variances for Interest in Animation (Teachers)

Levene Statistic	df1	Df2	Sig.
1.879	2	54	.163

Table 15: Test of Homogeneity of Variances for Interest in Games (Teachers)

Levene Statistic	df1	Df2	Sig.
.760	2	54	.473

Neither the results obtained from interest in creating animation nor interest in creating games were normally distributed ($\alpha = 0.05$). See Tables 12 & 13.

The assumption for homogeneity of variance was accomplished for the interest in animation sample and interest in games sample ($\alpha = 0.05$). See Tables 14 & 15.

Despite the fact that some of the data were not normally distributed, ANOVA was used for both data sets (interest in animation and interest in games), according to Steckler & Oleson (2005) ANOVA is robust enough over moderate violations of the distribution assumption.

One-way ANOVA was conducted to examine whether teachers' interest for creating animations was stronger after the summer institute ended compared to the beginning of the summer institute, and whether teachers' interest for creating animations had faded, increased or decreased two months after the institute ended. The results obtained are shown on Table 16.

Table 16: ANOVA for Interest in Creating Animations (Teachers)

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	8.066	2	4.033	4.313	.018
Within Groups	50.495	54	.935		
Total	58.561	56			

A separate one-way ANOVA was also conducted to examine teachers' interest in creating computer games. The results are shown in Table 17.

Table 17: ANOVA for Interest in Creating Computer Games (Teachers)

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3.590	2	1.795	1.636	.204
Within Groups	59.252	54	1.097		
Total	62.842	56			

Because a significant difference was found for teachers' interest in creating animations a planned contrast analysis was conducted. See Tables 18 and 19.

Table 18: Contrast Coefficients for Interest in Animations

	SurveyNum		
Contrast	1.00	2.00	3.00
1	-2	1	1
2	0	-1	1

Table 19: Contrast Tests for Interest in Animations

	Contrast	Value of Contrast	Std. Error	t	df	Sig. (2-tailed)
Assume equal variances	1	-.5335	.52695	-1.012	54	.316
	2	-.9211	.32811	-2.807	54	.007

There was a significant effect on participants' interest in creating animation, $F(2,54)=.02$, $\rho <.05$.

Planned contrasts revealed that teachers' interest for creating animations significantly decreased between the end of the summer camp and two months after the summer camp ended, $t(54)= 2.81$, $\rho <.05$, $r=.36$. There was no significant effect in teachers' interest in creating animation between the beginning of the summer institute and at the end of the summer institute, $t(54) = 1.01$, $\rho <.05$.

Teachers were asked after the summer institute ended if they were planning working on their animations in the future. Nineteen teachers out of twenty two responded the survey; the results are shown in Figure 9.

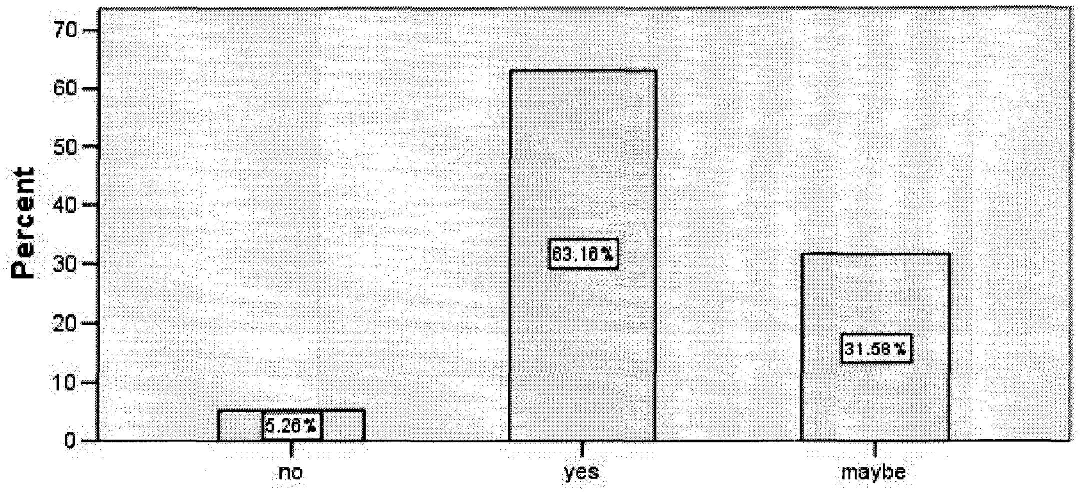


Figure 9: Are you planning on continuing working on your project after the teacher institute ends?

The results from the survey showed that 5.26 % did not plan working on their projects anymore, 63.16% planned continuing working on their project, and 31.58% thought they might work on their project.

Two months after the summer institute ended teachers were asked in a survey if they indeed worked in their animation and game projects after the summer institute ended and the responses are shown on Figure 10. Sixteen teachers out of twenty two responded the survey.

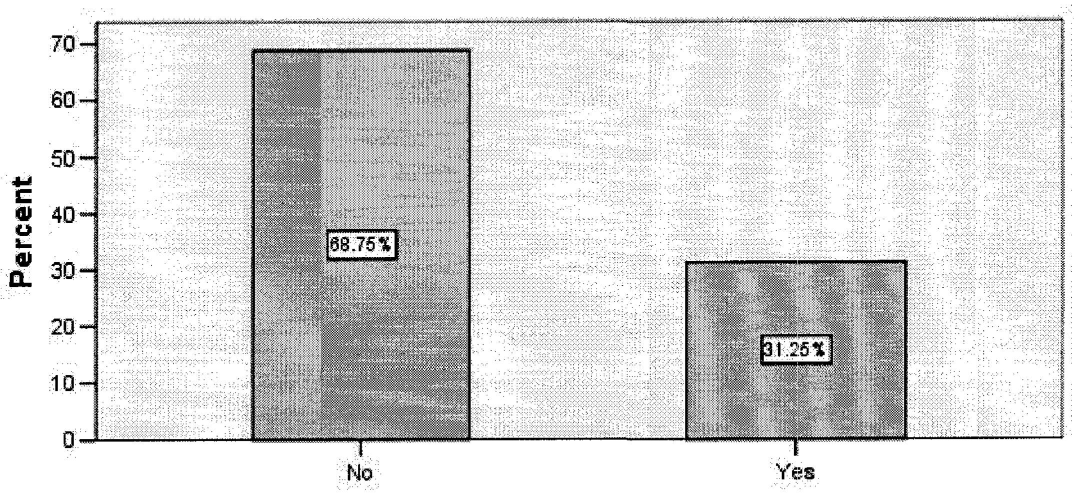


Figure 10: Did you keep working in the final project you created in the summer institute 2007 (two months post institute)?

The results showed 68.75% teachers did not continue working in their final projects while 31.25 % teachers did continue working on their projects.

Finally, participants were asked in the same follow up survey if they had created any new animation or game after the summer institute ended. See Figure 11.

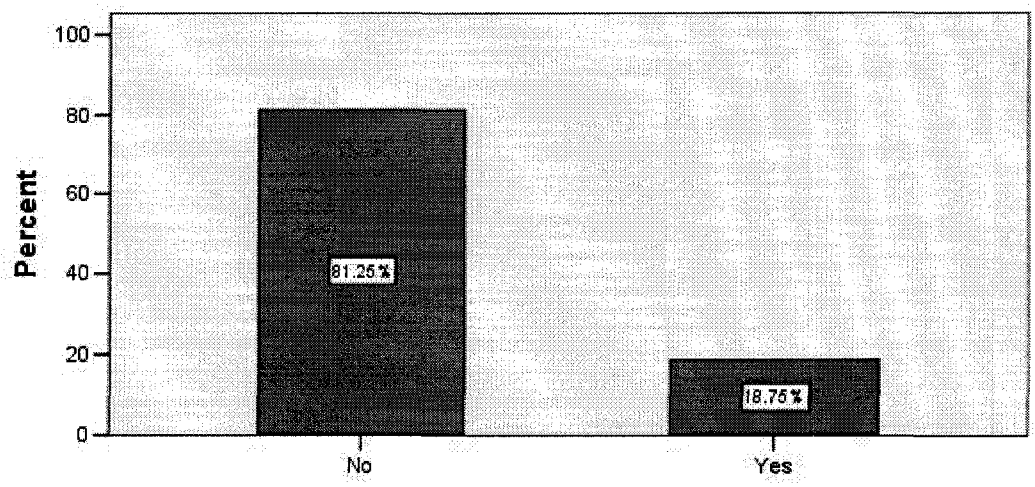


Figure 11: Have you created any new game or animation after the institute ended?

A majority of 81.25% of the teachers did not create any new animation or game, and 18.75% created one.

(b) How do teachers' previous experiences affect their interest in programming and task performance when creating interactive software?

To answer this question, results obtained from a programming attitudes survey, pre-summer institute survey, and subjective ratings for programming task 1 were analyzed using Pearson correlation. Two subjects were excluded from this analysis because of their high degree of computer expertise and programming experience.

The programming survey was given a few days before the summer institute ended, and it was the same as the one given to the students.

The subjective scores for the programming task 1 were based on the same two criteria used for rating students' task 1; a) how well they understood the computer coordinate system to properly place objects on the screen, and b) how well they understood the order to display objects on the screen using programming.

From the pre- summer institute survey, teachers were asked to rate how strong they thought their knowledge about using computers was. See Table 20.

Table 20: Pearson correlation for teachers' programming attitudes, performance on task, knowledge about computers and perceived self-efficacy towards math (significant results were highlighted)

	1	2	3	4	5	6	7	8	9	10
Teachers (N=20)										
1=comfortable	--	.46	-.70	.88	.35	.59	-.51	.86	.37	.75
2=interesting		--	-.32	.36	.93	.15	-.12	.45	.47	.41
3=challenging			--	-.70	-.19	-.83	.42	-.64	-.60	-.43
4=good at				--	.31	.51	-.51	.61	.33	.73
5=excited					--	.00	.00	.34	.30	.37
6=good in task						--	-.46	.52	.64	.41
7= trouble							--	-.46	-.62	-.53
8=easy to								--	.44	.57
9=Task1									--	.43
10=Computers										--

A positive relationship was found between how comfortable teachers were doing programming and how interesting they thought programming was, $r=.46$, $\rho <.05$. However, they did not think they were comfortable doing programming when it was perceived as very challenging, $r=-.70$, $\rho <.01$, or when they had a lot of trouble with the programming tasks, $r=-.51$, $\rho <.05$.

Moreover, participants felt most comfortable with programming when they felt that they were good at programming, $r=.44$, $\rho <.01$, they were doing well in the programming tasks, $r=.59$, $\rho <.05$, and they felt the programming was easy to do, $r=.86$, $\rho <.01$. Having strong knowledge about using computers was also positively correlated with being comfortable with programming, $r=.75$, $\rho <.01$.

Finding programming to be an interesting task was positively correlated with being excited about programming, $r=.93$, $\rho <.01$, while being good at programming was negatively correlated with the amount of perceived challenge from doing programming, $r=-.70$, $\rho <.05$.

Perceiving programming to be a challenging task was also negatively correlated with being good in the programming tasks, $r=-.83$, $\rho <.01$, negatively correlated with how easy programming was, $r=-.64$, $\rho <.01$, and negatively correlated with performance on task 1, $r=-.60$, $\rho <.01$.

Being good at programming was positive correlated with; a) being good at the programming task, $r=.51$, $\rho <.05$, b) easy to learn programming, $r=.61$, $\rho <.01$, and c) strong knowledge on using computers, $r=.73$. Good at programming was negatively correlated with having trouble with the programming task, $r= -.51$, $\rho <.05$.

A negative relationship was found between doing well in the programming task and how much trouble teachers had with programming, $r=-.46$, $\rho <.05$. Conversely, a positive correlation was found between doing well in the programming tasks and how easy it was to do programming, $r=.52$, $\rho <.05$, as well as between performance on task 1 and perceptions of doing well in the programming task, $r=.64$, $\rho <.01$.

Having trouble understanding things related to programming was negatively correlated with; a) perceiving programming as easy to learn, $r= -.46$, $\rho <.05$, b) performance on task 1, $r= -.62$, $\rho <.01$, and c) knowledge about using computers, $r=-.53$, $\rho <.05$.

How easy learning to program was for the teachers was positively correlated with strong knowledge about how to use computers, $r=. 57$, $\rho < .05$.

c) What factors influenced teachers' interest for creating games and animations?

This question will be addressed in chapter 5 using qualitative analysis.

Research Question 3: What are the differences and similarities between students' and teachers' levels of interest throughout the process of creating games and animations?

To investigate whether there were any differences between students' and teachers' interest for creating animations, a mixture of between-group (students and teachers) and repeated measures analysis of variance was conducted to compare changes of interest ratings for creating games and for creating animations among students and teachers at the beginning of the workshop, the end of the workshop and two months after the workshops ended.

Assumptions for conducting analysis of variance were first checked. The results for the test of normality, homogeneity of variance and sphericity are shown in the, Tables 21, 22, and 23.

Table 21: Tests of Normality

Type of Participant		Kolmogorov-Smirnov(a)			Shapiro-Wilk		
		Statistic	df	Sig.	Statistic	df	Sig.
Pre-Interest Animation	teacher	.288	15	.002	.783	15	.002
	student	.232	11	.100	.822	11	.018
Post-Interest Animation	teacher	.353	15	.000	.728	15	.001
	student	.232	11	.100	.822	11	.018
Follow Up-Interest Animation	teacher	.173	15	.200(*)	.876	15	.042
	student	.277	11	.018	.799	11	.009

* This is a lower bound of the true significance.

Table 22: Test of Homogeneity of Variance for Animations (Students & Teachers)

		Levene Statistic	df1	df2	Sig.
Pre-Interest Games	Based on Mean	.019	1	24	.890
Post-Interest Games	Based on Mean	.154	1	24	.699
Follow-Interest Games	Based on Mean	.059	1	24	.809

Table 23: Mauchly's Test of Sphericity(b) (Students & Teachers)

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	Df	Sig.	Epsilon(a)		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
Interest in animation	.629	10.661	2	.005	.729	.797	.500

Some of the data were not normally distributed ($\alpha = 0.05$) but because ANOVA is robust enough over moderate violations of the distribution assumption (Steckler & Oleson, 2005) this method was used.

The assumption of homogeneity of variance was accomplished, $\rho > .05$, and the assumption of sphericity was not confirmed, so the Greenhouse-Geisser estimator was used in the analysis. Descriptive statistics are shown on Tables 24 and 25.

Table 24: Within-Subjects Factors for Animations (Students & Teachers)

Interest animation	Dependent Variable
1	Pre-Interest Ani
2	Post-Interest Ani
3	Follow Up- Interest Ani

Table 25: Descriptive Statistics for Animations (Students & Teachers)

	Type of Participant	Mean	Std. Deviation	N
Pre-Interest Animation	Teacher	4.3333	.72375	15
	Student	3.8182	.75076	11
	Total	4.1154	.76561	26
Post-Interest Animation	Teacher	4.3333	.97590	15
	Student	3.8182	.75076	11
	Total	4.1154	.90893	26
FollowUp-Interest Animation	Teacher	3.6000	1.12122	15
	Student	3.7273	.78625	11
	Total	3.6538	.97744	26

Mauchly's test indicated that the assumption of sphericity had been violated ($\chi^2(2) = 10.66$, $\rho < .05$); therefore degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity (ε=.73). The results shown in Tables 26 and 27 indicate that the interest in creating animations was not significantly affected after participating in the workshops, $F(1.46, 35.01)=3.47$, $\rho > .05$.

There was no significant interaction effect between interest in creating animation over time and the type of participant (student or teacher), $F(1.46, 35.01)= 2.11$, $\rho > .05$.

There was a non-significant (ns) main effect between students' and teachers' interest in creating animations (type of participant) over time, $F(1, 24) < 1$, ns.

Table 26: Tests of Within-Subjects Effects for Animations (Students & Teachers)

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Interest ani	Sphericity Assumed	2.874	2	1.437	3.467	.039
	Greenhouse-Geisser	2.874	1.459	1.970	3.467	.056
Interest ani * TypeParticipant	Sphericity Assumed	1.746	2	.873	2.106	.133
	Greenhouse-Geisser	1.746	1.459	1.197	2.106	.148
Error(interest ani)	Sphericity Assumed	19.895	48	.414		
	Greenhouse-Geisser	19.895	35.012	.568		

Table 27: Tests of Between-Subjects Effects Animations (Students & Teachers)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	393.737	1	393.737	791.294	.000
Type Participant	.575	1	.575	1.156	.293
Error	11.942	24	.498		

The same procedure to compare changes of interest in creating animations was conducted to study changes of interest in creating computer games. See Tables 28, 29, and 30.

Table 28: Tests of Normality for Games (Students & Teachers)

Type Participant		Kolmogorov-Smirnov(a)			Shapiro-Wilk		
		Statistic	df	Sig.	Statistic	df	Sig.
Pre-Interes Games	Teacher	.311	15	.000	.737	15	.001
	Student	.236	11	.089	.865	11	.066
Post-Interest Games	Teacher	.318	15	.000	.778	15	.002
	Student	.264	11	.031	.878	11	.099
Follow-Interest Games	Teacher	.214	15	.062	.859	15	.023
	Student	.228	11	.116	.916	11	.285

Table 29: Test of Homogeneity of Variance (Students & Teachers)

		Levene Statistic	df1	df2	Sig.
Pre-Interest Games	Based on Mean	.019	1	24	.890
Post-Interest Games	Based on Mean	.154	1	24	.699
Follow-Interest Games	Based on Mean	.059	1	24	.809

Table 30: Mauchly's Test of Sphericity (Students & Teachers)

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon(a)		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
Interest in games	.912	2.120	2	.346	.919	1.000	.500

Some of the data were not normally distributed ($\alpha = 0.05$) but because ANOVA is robust enough over moderate violations of the distribution assumption (Steckler & Oleson, 2005) this method was used.

The assumption of homogeneity of variance was accomplished, $\rho > .05$, and the assumption of sphericity was confirmed, $\rho > .05$. Descriptive statistics are shown in Tables 31 and 32.

Table 31: Within-Subjects Factors Games (Students & Teachers)

Interest animation	Dependent Variable
1	Pre-Interest Games
2	Post-Interest Games
3	Follow Up- Interest Games

Table 32: Descriptive Statistics Games (Students & Teachers)

Type Participant		Mean	Std. Deviation	N
Pre-Interes Games	Teachers	4.0667	1.22280	15
	Students	3.5455	1.21356	11
	Total	3.8462	1.22286	26
Post-Interest Games	Teachers	4.2000	1.01419	15
	Students	3.1818	1.32802	11
	Total	3.7692	1.24283	26
Follow-Interest Games	Teachers	3.7333	1.09978	15
	Students	3.2727	1.19087	11
	Total	3.5385	1.13950	26

The results showed interest in creating computer games was not significantly affected by participating in the workshops, $F(2, 48)=1.25$, $\rho > .05$. See Table 33.

There was no significant interaction effect between the interest in creating games over time and the type of participant (student or teacher), $F(2, 48)= 1.25$, $\rho > .05$. See Table 33.

There was a non-significant (ns) main effect between students' and teachers' interest in creating games (type of participant) over time, $F(1, 24) < 1$, ns. See Table 34.

Table 33: Tests of Within-Subjects Effects Games (Students & Teachers)

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
Interest in games	Sphericity Assumed	1.188	2	.594	1.250	.296
Interest in games * Type Participant	Sphericity Assumed	1.188	2	.594	1.250	.296
Error (interest in game)	Sphericity Assumed	22.812	48	.475		

Table 34: Tests of Between-Subjects Effects for Games (Students & Teachers)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	1023.846	1	1023.846	323.320	.000
Type Participant	8.462	1	8.462	2.672	.115
Error	76.000	24	3.167		

A series of independent t-tests were also implemented to explore differences between students' and teachers' attitudes towards programming, performance on tasks, and previous experience.

The first t-test compared students' and teachers' ratings of previous knowledge about using computers, these results were obtained in the first survey they completed once the summer workshop started. Scores went from 1 to 5, where 1 meant they had very little knowledge about how to use computers and 5 meant they had strong knowledge about using computers. See Table 35.

Table 35: Group Statistics for Experience with Computers

Type of Participant		N	Mean	Std. Deviation	Std. Error Mean
Exp Compu	Teachers	19	3.6316	.83070	.19058
	Students	12	3.4167	1.16450	.33616

On average, high school teachers rated their knowledge about using computers higher ($M = 3.63$, $SE = .19$), than high school students ($M = 3.42$, $SE = .34$). This difference was not significant $t(29) = .60$, $\rho > .05$, r= .11. See Table 36.

Table 36: Independent Samples Test for Experience with Computers

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	T	df	Sig. (2-tailed)	Mean Differ	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Comp	Equal variances assumed	1.23	.28	.60	29	.55	.21	.36	-.52	.95
	Equal variances not assumed			.56	18.07	.59	.21	.39	-.60	1.03

The second independent t-test evaluated whether students perceived themselves as better artists than teachers. These results were obtained by the pre-survey, participants were asked to rate how good they think they were at drawing. Ratings came in a 1-5 scale, where 1 meant low perceived ability for drawing and 5 was high ability. This question was included because drawing is an essential skill to create animations. See Table 37.

Table 37: Group Statistics for Being Good at Drawing

Type of Participant		N	Mean	Std. Deviation	Std. Error Mean
Art	teacher	19	1.8947	1.10024	.25241
	student	12	3.5000	1.08711	.31382

On average, high school teachers rated their ability for drawing lower (M = 1.89, SE = .25), than high school students (M = 3.50, SE = .31). This difference was significant $t(29) = -3.98$, $\rho < .05$, r= . 59. See Table 38.

Table 38: Independent Samples Test for Art Skills

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Differ	Std. Error Differ	95% Confidence Interval of the Difference	
									Lower	Upper
Art	Equal variances assumed	.017	.90	-3.98	29	.000	-1.61	.40	-2.43	-.78
	Equal variances not assumed			-3.99	23.76	.001	-1.61	.40	-2.43	-.77

The third one-tail t-test compared if students performed better on the first programming task than the teachers, based on observations from the camp instructor, this seemed to be an expected result because young students are usually more comfortable about learning new technologies. See Table 39.

Table 39: Group Statistics for Performance on Programming Task 1

Type of Participant		N	Mean	Std. Deviation	Std. Error Mean
Task1	teacher	19	3.2632	1.19453	.27404
	student	12	4.0000	.95346	.27524

On average, high school students perform better in the programming task # 1 ($M = 4.00$, $SE = .28$), than high school teachers ($M = 3.26$, $SE = .27$). This difference was significant $t(29) = -1.8$, $\rho < .05$, r= . 31. See Table 40.

Table 40: Independent Samples Test for Programming Task 1

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Task1	Equal variances assumed	1.13	.30	-1.8	29	.082	-.74	.41	-1.57	.10
	Equal variances not assumed			-1.9	27.25	.068	-.75	.39	-1.53	.06

The last set of t-tests contrasted students' and teachers' self-efficacy towards programming, perceived effort towards programming, and positive attitudes. See Table 41.

Table 41: Group Statistics for Programming Survey

Type of Participant		N	Mean	Std. Deviation	Std. Error Mean
Q1=comfortable	teacher	20	2.8500	1.08942	.24360
	student	12	3.4167	.79296	.22891
Q2=interesting	teacher	19	4.1579	.95819	.21982
	student	12	4.0000	.95346	.27524
Q3=challenging	teacher	20	4.2500	.85070	.19022
	student	12	3.0833	.99620	.28758
Q4=good at	teacher	20	2.3500	1.03999	.23255
	student	12	3.1667	.71774	.20719
Q5=excited	teacher	20	4.0000	.97333	.21764
	student	12	3.5000	.79772	.23028
Q7=good in task	teacher	20	3.4500	.68633	.15347
	student	12	3.7500	.96531	.27866
Q8= trouble	teacher	20	2.6500	1.13671	.25418
	student	12	2.2500	.86603	.25000
Q9=easy to	teacher	20	2.9500	.94451	.21120
	student	12	3.5000	.90453	.26112

The results from the programming survey showed that only two categories were significantly different between students and teachers, "challenging" and "good at programming". The results from "challenging" were obtained from survey results on a 1-5 point scale, where 1 meant low "challenge" to do programming, and 5 meant very "challenging" to do programming.

On average, high school teachers rated the amount of challenge to do programming higher (M = 4.25, SE = .19), than high school students (M = 3.08, SE = .29). This difference was significant $t(30) = 3.52$, $\rho < .05$, r= .54. See Table 42.

Results to address the question of how able participants felt they were at programming were obtained from the programming survey on a 1-5 scale, where 1 meant not so good, and 5 meant very good.

On average, high school students perceived themselves as better programmers (M = 3.17, SE = .23), than the teachers (M = 2.35, SE = .23). This difference was significant $t(30) = -2.39$, $\rho < .05$, r= .39. See Table 42.

Table 42: Independent Samples Test for Programming Survey (significant results were highlighted)

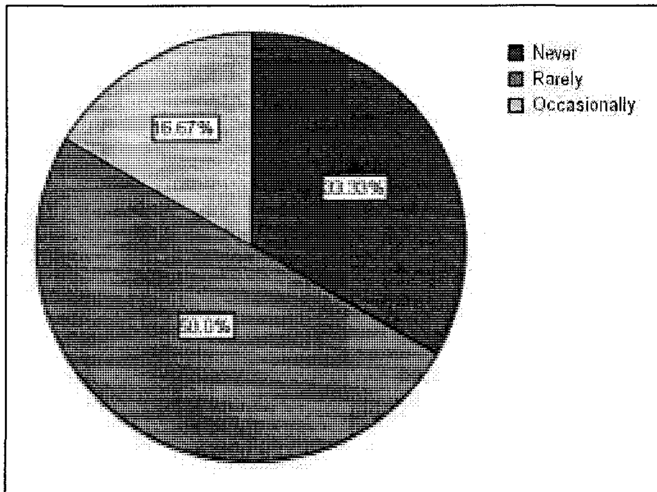
		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Q1=comfortable	Equal variances assumed	1.08	.31	-1.57	30	.13	-.57	.36	-1.31	.17
Q2=interesting	Equal variances assumed	.40	.53	.45	29	.66	.16	.35	-.56	.88
Q3=challenging	Equal variances assumed	.21	.65	3.52	30	.001	1.17	.33	.49	1.84
Q4=good at	Equal variances assumed	3.57	.07	-2.39	30	.02	-.82	.34	-1.51	-.12
Q5=excited	Equal variances assumed	.58	.45	1.50	30	.14	.50	.33	-.18	1.18
Q7=good in tasks	Equal variances assumed	1.89	.18	-1.03	30	.31	-.30	.29	-.90	.30
Q8= trouble	Equal variances assumed	.71	.41	1.05	30	.30	.40	.38	-.38	1.18
Q9=easy to	Equal variances assumed	.14	.71	-1.62	30	.12	-.55	.34	-1.24	.14

Research Question 4: What kind of games or animations did students and teachers create as a final project? How different was the process of creating the final project for students and teachers?

A qualitative analysis is presented in chapter 5 to address this question.

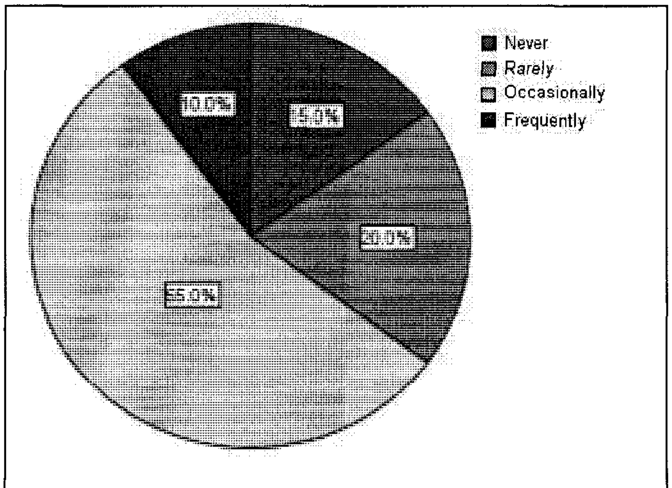
Research Question 5: Did teachers and students use the class web site and online community?

Participants were asked in the post-summer workshop survey how often they used the class website. The results from participants can be found on Figure 12 and 13.



Students' responses

Figure 12: How often did you use the class website? (Students' responses)

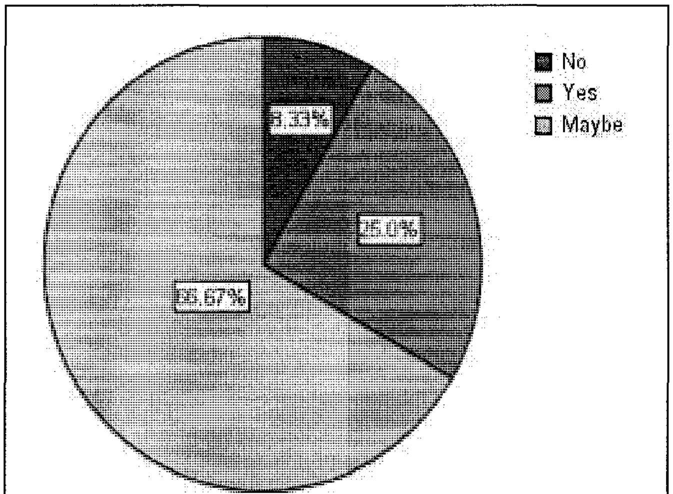


Teachers' responses

Figure 13: How often did you use the class website? (Teachers' responses)

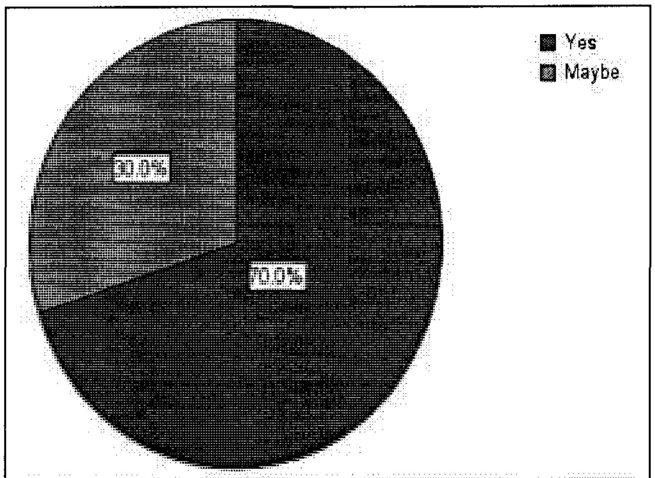
Teachers used the class website more often than students did; moreover 55 % of teachers indicated they consulted the website occasionally in contrast to 16.67% of students, who mentioned they used the website occasionally. Among 20% teachers used the class website rarely while 50% students chose this answer, 15% teachers never used the class website, whereas 33.33% students never used the class website. Finally 10% teachers used the class website frequently.

Participants were also asked in the post-summer workshop survey if they were planning to use the class website in the future. The results from both types of participants can be found on Figure 14 and 15.



Students' responses

Figure 14: Will you be using the class website in the future? (Students' responses)

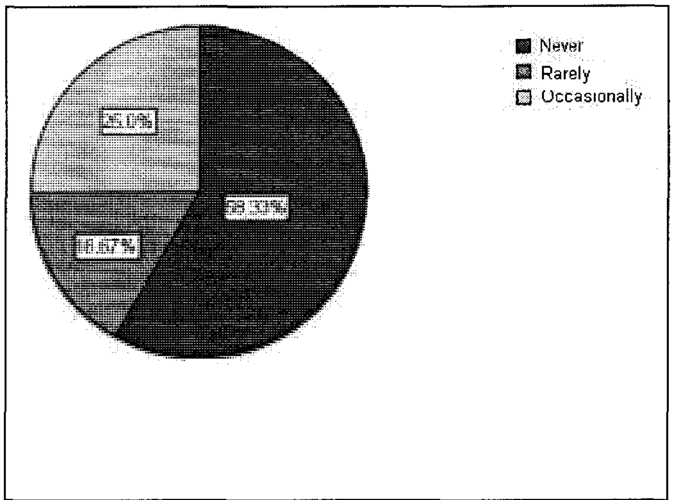


Teachers' responses

Figure 15: Will you be using the class website in the future? (Teachers' responses)

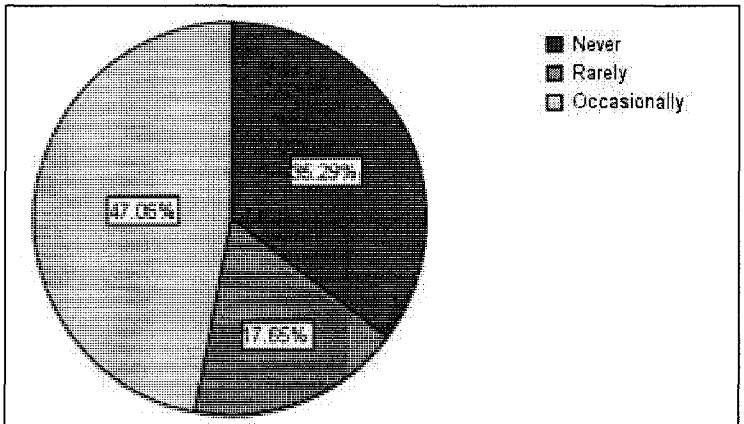
Only 8.33% students did not plan to use the website at all, 66.67% mentioned they might use the website in the future compared to 70% teachers who mentioned that they might use the website. Finally, 25% students plan on using the website in the future, compared to 30% of teachers who plan on using the website.

In a follow-up survey conducted two months after the summer workshops ended, participants were asked how often they had used the class websites. The answers to these questions are presented on Figure 16 and 17.



Students' responses

Figure 16: How often have you used the class website in the past two months? (Students' responses)



Teachers' responses

Figure 17: How often have you used the class website in the past two months? (Teachers' responses)

Teachers used the class website more often than the student participants. Moreover, 47.06% of teachers used it occasionally while only 25% of students used the website, 17.65% of teachers used the website rarely compared to 16.67% of students. Finally 35.29% teachers never used the website while 58.33% of students never used the class website.

Chapter 5 presents qualitative analysis of students' and teachers' data on the creation of games and animations. Descriptions about what they did for their programming tasks and what they said about their favorite activities in the camp are included. Additionally, final projects screenshots are displayed and categorized, and website patterns of use and preferences are discussed.

Chapter 5: Learning Environment and Participants' Work

The learning environment is crucial to support learners' interest, and for this reason an explanation of the environment, the cycle of work, and the resources available for the participants of the workshops are included in this chapter to help readers further understand interactions among participants and their learning environment.

To better understand participants' interest in creating games and animations, a qualitative approach was taken. Participants' work progress was collected and analyzed according to what they were taught to do for each assignment; also, factors that might influence participants' interest were identified and included in the analysis. Final projects were analyzed for both students and teachers, and comparisons were made.

This chapter ends with an analysis of the use of class website and online community by each group of participants.

Learning Environment

All the learning activities associated with the summer camp and teacher institute were listed by day in Appendix F.

The learning environment for students and teachers was analyzed by coding the activities conducted in the camp using self-determination theory and the cognitive apprenticeship/ studio thinking perspective. Field notes and instructors' interviews were coded to reflect characteristics that are relevant to these theories of learning. Self-determination characteristics were observed by looking at opportunities that promote a sense of *relatedness*, *autonomy*, and *competence* on participants.

Relatedness was shown in those activities in which participants shared their knowledge and experiences with the members of the group, and activities that showed how supportive other people were about their ideas and personality. *Autonomy* was manifested when participants were allowed to make decisions, and *sense of competence* was found in activities or behaviors that informed participants about how well they were accomplishing the assigned tasks.

"Cognitive apprenticeship" and "studio thinking" habits were coded using five categories; modeling the task, scaffolding the activities, fading the amount of help, coaching the final project, emphasizing the relevance of the task, and reflecting on the learning process.

Self-determination Characteristics

The researcher collected examples of activities that influence participants' self-determination by taking notes of all the activities that happened in the workshops every day, and then she looked for activities that shared characteristics with each of the elements that define self-determination theory, i.e., sense of relatedness, autonomy, and sense of competence.

Sense of Relatedness:

Instances of relatedness for students came particularly at the beginning of the summer camp and towards the end of the summer camp.

The first day of the camp, the instructor asked students to play an ice breaking game, the game consisted of introducing another person to the class by mentioning two things that were true about the person, and one thing that was a lie (these things were discussed previously between pairs of participants), the class had to guess what the lie was about the person. Although there were good intentions behind the use of the game, it took more than an hour to get everybody introduced this way and differences among students were highlighted, especially differences related to economic status and skill levels. For instance, one participant mentioned that she had five horses and this was not a lie; another participant mentioned that she had travelled almost all the world and spoke several languages and none of those were lies; another girl mentioned that she was web moderator for three websites; another student mentioned he had never travelled outside Washington state; another student mentioned she shared a house with eight people.

The intent of this activity was to provide a sense of belonging to an environment by finding similarities among the peers. However, as it unfolded in this session, this activity did not provide a good sense of relatedness for the students because of the emphasis on differences instead of similarities; therefore social interactions and collaboration were limited in the summer camp.

Teacher participants also had the same introductory game but the instructor decided to play it just for half of the class because of time constraints, so some of the teachers did not have a chance to introduce themselves to the class by saying two things that were true about themselves and one lie. However, overall this introductory game was more appropriate for the teachers as compared to the student group, since teachers tried to say humorous things about themselves instead of emphasizing social status or skill levels.

Lack of diversity in the student group in terms of ethnicity seemed to play a role in the sense of relatedness, in particular, for the only African-American student enrolled in the summer camp. The second day after the summer camp started the student mentioned to a summer camp organizer that she felt as she did not have anyone to talk to in the camp, so the next day she decided to bring a friend to the camp without asking for permission, which caused a little confusion among the camp members. The instructor accepted her friend in the camp, and she started working on the programming tasks along with the former camp member. The friend was there to be a companion, so she started to become a distraction for her camp member friend. At the end, the guest friend did not create any final project, and the summer camp student got behind on her final project which she had to finish by herself.

The teacher group was very homogeneous in term of ethnicity but there were a lot of differences in terms of skill levels: there were two professional programmers in the group with high level skills, and there were also teachers who had limited experience with computers. Because of that, one of the teachers who appeared to be having problems getting the programming assignments done got singled out by one of the program organizers and the teacher received some discouraging words from the summer institute organizer. At the end, the teacher paired with a more skillful teacher and continued working on the programming tasks. He received help from instructors and TAs and was able to finish the summer institute successfully.

Sense of relatedness was also perceived through the help students and teachers received from the teaching assistants (TAs) and instructors. There were five TAs and at least one instructor at any given time and they provided as much help and guidance as possible to the participants. One student participant mentioned in the survey that one of his favorite parts of the camp was being around the TAs.

In general, the camp activities did not promote much collaboration among participants because most of the initial tasks were accomplished individually or with the help of the TAs and instructor. Also, there was pressure to get things done quickly so the instructor could move on with the next lesson. In contrast to students, from the researcher's observation teachers collaborated more with each other and they did not need any encouragement to do so.

The programming and Photoshop learning activities, which consisted of basic instruction about using graphics programming with Visual Basic and learning to draw and edit images in Photoshop, were very academically oriented most of the time. One instructor taught the lesson and the rest of the participants observed and followed the steps to do the task. Only two activities promoted some social interaction for the students group; one was creating a PowerPoint presentation with a designated partner to show how animations were used in different fields, and the other one was creating the clay model to do a claymation. Particularly the last activity provided a relaxed environment for the students to sit together around two big tables in the school hallway to create their models, share the materials, and talk about games they liked to play and other activities.

Despite the class instruction being the same for students and teachers, teachers stopped the instructor frequently to ask him questions about what he was doing, whereas students tended to just follow what the instructor did.

Towards the end of the camp, when students had the option to work in pairs for their final project, only a few of them were willing to work together, so the TAs and instructor had to encourage the students to work together so that they could be more efficient with their project and present a finished product at the end. On the other hand, teachers were more willing to work in pairs than students; part of this was because several teachers enrolled in the summer institute with their coworkers and they already knew they were going to work together.

The last instance of relatedness was observed the last day of the summer camp, when students had to do a public presentation of their final project in front of the parents and relatives who were invited to the final presentation. Students were very nervous about the public presentation with the parents; this was particular true for students whose performance during the camp had been less proficient. Moreover, several students refused to give their final project to the TAs when they were asked to do it because they wanted to make it better and present a more polished product. One of the low proficiency students was very nervous about the presentation because she did not have time to finish or prepare what she wanted to say, so when the TAs were distracted she wrote her name at the beginning of the list that represented the order of presenters, then she did her presentation and left the camp.

Teachers' final presentations were done only in the presence of their summer institute classmates so the social dynamics were different from the students' dynamics for the final presentation. Also, teachers had written guidelines with specific things they were asked to present and mention in their presentation, while the students' presentation were less structured.

Autonomy:

Although participants had specific programming tasks to accomplish, they were provided with choices about the type of art they wanted to create and the types of stories or learning tasks they wanted to include in their animations or games. Also, for the final project students had the freedom to create a game or an animation, while the teachers were expected to create educational software that taught some principle about a subject they teach in their schools.

During, the last two days of class, students had a couple of hours to focus on the activity they needed to accomplish for their final project, art work or programming. In contrast, teachers had one week devoted to work on their final project.

The sense of autonomy reflected in how much control participants had over the programming was somewhat limited because they did not receive any type of explanation about the function of the compiler and how to use it to find errors in their programming, so participants, in particular students, depended entirely on the TAs to debug their programs.

There was also a lack of understanding about the file structure created by Visual Basic to store the programming code; this caused participants to lose their programming code or to have several unrelated programming files stored in the same project folder.

Additionally, students did not have written material to see the code example presented by the instructor when he was teaching, especially during the first week of class. Students needed to rely only on what they saw on the projection screen. Several students in the back of the classroom had a hard time trying to read the letters projected on the board from the distance and they were lost trying to follow the code. The printouts of the code were given to students at the end of the class and sometimes multiple printouts were given to them and they were stored in their bags but they did not use them at all during class. They got a book with the code at the end of the class but at that point, they had already finished the final project.

Teachers were given a user manual book with all the code at the beginning of the summer institute and they could always refer to their user manual when they got lost in the programming tasks. Additionally several teachers bought Visual Basic books to complement what they were learning in the summer institute.

Sense of Competence:

The instructor showed animations and educational software from previous years so participants could get a sense of what previous high school students and teachers accomplished with the skills they learned and the time they had. The problem was that the animations and educational software showed by the instructor were mostly the best animations and games developed by the best participants of the previous year. Some of these animations seemed to be created by participants who got lots of help, or had previous knowledge about using this type of technology; they might had spent a lot of time outside the camp working on their animations. From just looking at the animations it was hard to have a real sense of how much time was spent on the animation. For participants who demonstrate having difficulty accomplishing the programming tasks this could cause them to feel hopeless about the expectations the instructor had on them.

Teachers expressed more concern about their abilities to do programming and stressed out more about technical problems than the students. Some teachers experienced several frustrations because they lost their programming code several times due to lack of understanding about what files they needed to save in their personal thumb drives, or because the temp file in their computer got deleted once the machines went into sleep mode during lunch time and that was the only place in their computers where they could save their programs.

Sense of competence for several teachers was low because they had to rely a lot on the instructor and TAs to complete their final project, the final project programming load was beyond their abilities, and they did not realize these things until it was too late so they had to request substantial help.

Cognitive Apprenticeship Characteristics

Some characteristics of "cognitive apprenticeship" and "studio thinking" were observed in the workshops. Modeling the task using direct instruction was the main form of teaching during the workshop. Participants observed what the instructor did, and immediately copied the programming instructions and observed the results. If participants could not achieve the same results as the instructors, the TAs helped them with the code until it worked.

Scaffolding the activities was observed at a later point in the workshop, when the participants were working on their final project. Because of the short time they had to finish their projects (in particular the students), several features of cognitive apprenticeship were not implemented: fading the amount of help was not possible, because students and teachers were in a rush to finish their projects to do the final presentation. Also, reflecting on what they learned and learning how to evaluate and criticize other animations or games were not included as part of the instruction process.

In terms of the relevance of the task and immediacy in applying the skill they learned, participants had plenty of opportunities to see the relevance and use of the skills. The instructor demonstrated and exemplified the use of a math equation and process, then immediately he incorporated that into the animation to show how it worked, followed by participants' use of the same technique in their programs.

Also participants used professional tools and applications to create their game or animations, such as Microsoft Visual Basic, Adobe Photoshop, and tablets. This is one of the main features of a cognitive apprenticeship and studio learning environment.

Participants' Work

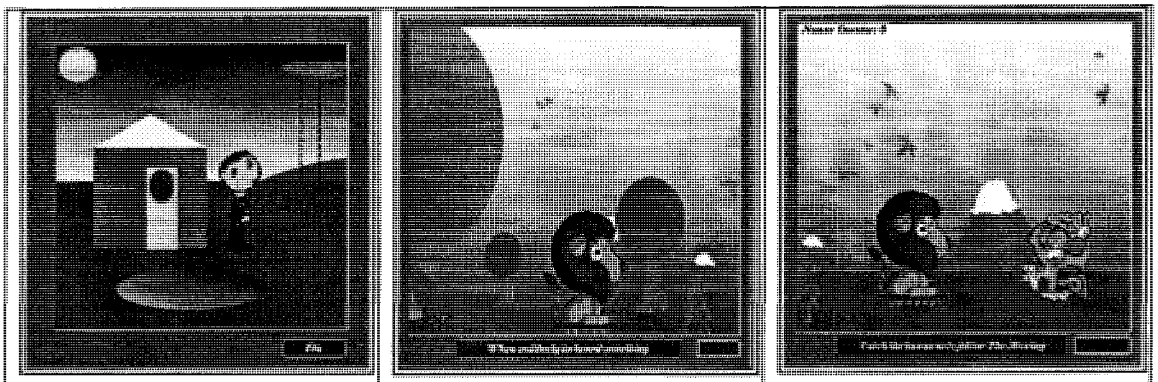
Six case studies were chosen to observe participants' progress, and analyze how their interest fluctuated during their experiences in the workshops. These case studies represent a sample of individuals with different level of skills and motivations to participate in the workshops. Also these case studies include different strategies participants used to finish their programming tasks.

Participants' work and interest were analyzed based on interest theory. Answers from the surveys, class observations and programming tasks were categorized according to participants' previous knowledge, positive feelings towards games and animations, interest to join the camp, self-efficacy for programming, performance on task, perceive effort, persistence and resource use over time.

Students' Work

The first student was Billy. He reflected the progress of a student who was interested in creating games but did not understand the level of challenge involved in creating games. He wanted to create a baseball game at the beginning and spent several hours working on his characters and drawing the body position for the baseball players, but later on he realized the complications involved in creating a sport game and changed his original idea to a simpler game he was able to finish. Figure 18 show a screenshot of each of Billy's programming tasks and the number of lines of code he wrote for each task.

Billy:



Programming Task #1	Programming Task # 2	Programming Task # 3
#Lines Code: 55	#Lines Code: 110	#Lines Code: 271

Figure 18: Billy's Programming Tasks

Billy was a 10th grader who had not created any type of computer animation or computer game before, but who had taken classes to learn different 3D modeling software, such as AutoCAD and Rhino. His previous knowledge about creating games was stronger than the average student in the summer camp.

He had strong positive attitudes towards computers and he mentioned that he enjoyed working with computers a lot. Drawing and watching animations were rated as neutral activities by him, but playing computer and video games was rated as his favorite activity. He mentioned his favorite video/computer game was Major League Baseball 2k7.

In terms of his interest in the summer camp and for creating games and animations, he mentioned that what attracted him the most about the summer camp was the ability to work on computers. At the beginning of the camp he had a fairly strong interest in creating games and animations and he rated his interest as the same for both of them.

This student came to the camp with a strong self-efficacy for how to use computers, and a strong background and self-efficacy for doing math; however he rated his drawing skills as average.

As far as his performance for programming task # 1, which consisted of creating an scene that included a house, lake, clouds, sun and rain using geometric shapes, he demonstrated a good understanding of the computer coordinate system to display graphics on the screen. To create a composition of a scene, he got a house on the ground, his character was aligned with the house, the door was located in the middle of the main house's wall, he added a window to the door, positioned the sun, and the lake and clouds correctly on the scene. He also demonstrated a good understanding of the order to place objects on the screen, for example, the clouds were in front of the sun, he positioned a hill in the back of the scene, the rain was falling down in the distance and drops were touching the ground before disappearing.

Billy's performance in the programming task 2, which consisted of creating a game-like program that contained a background image scrolling effect, a character who responds to user's keyboard input, some sound, and a storyline, was outstanding. He implemented a very smooth and fluid background scrolling; was successful at integrating music and sound effects; started working on a sort of storyline; added user inputs to control the character on the screen using the keyboard; and used timers effectively to change the storyline and to include a sound effect.

For the final project, Billy worked alone and elaborated his task # 2 further, adding a monkey character and a game play for the lion to eat bananas in the game and get points that were reflected in the score section of the screen. In the game when the lion missed a banana a monkey appeared on the screen and make fun of the lion. He added and introduction screen and instructions to the game, as well as a feedback screen and a final screen. He used mathematical equations to calculate distance between objects to make the bananas disappear, and show the monkey, and added an event to make the lion roar when the user pressed the "r" key.

The post-survey showed that Billy had very positive attitudes and self-efficacy towards programming, he found himself very comfortable with the programming activities and perceived himself as doing excellently in the programming tasks, also programming was easy to do for him. However he mentioned he was not that excited about programming. Billy's favorite activity was using Photoshop because he liked to edit images.

Despite Billy's positive attitude about the programming and using Photoshop, his interest for creating games and animations faded over time from being very interested at the beginning of the camp to being somewhat interested at the end of the camp and two months after the camp ended.

Sense of relatedness was not strong for this participant, he worked alone most of the time, he enjoyed the company of the other members of the camp only a little bit and he did not plan to keep in touch with anyone in the camp. In the follow-up survey he confirmed that he did not contact any of his camp classmates. However he mentioned that he had received a lot of encouragement from his school teachers to continue learning about technology.

He mentioned at the end of the summer camp that he might continue working on his project after the summer camp ended but in the follow up survey he mentioned that at the end he did not work on his game anymore and he had not created any new games or animations since the camp ended.

As for his career interest, he stated in the survey that the things he did in the camp didn't fit too much with want he wants to do as a career or what he does as a hobby.

This student mentioned in the survey that he worked at home occasionally on his final project and he used editing software called PhotoImpact to work on his project; he also received some help from family members and friends to complete his final project.

Billy was an example of a student who had lots of skills and self-efficacy for working with computers and was good at math. Mathematics and computers were two things that he enjoyed doing as well as playing computer games. He performed very well in all the programming tasks, and perceived himself as doing well with the programming assignments and understood programming very well. However, his interest in creating games and animations diminished over time from the beginning of the camp until two months after the camp. He was able to apply math in a meaningful way to his game final project. He was very familiar with the elements he needed in order to create a game, such as multisensory feedback (visual and sound), adding interactions, incorporating scores, adding a right amount of challenge, and adding humor and surprise. He had very good planning skills and he was able to modify his final project ideas on time so he could finish on time for the presentation.

From a motivation point of view, his interest, positive feelings, planning skills and pre-knowledge gave him the skills to succeed in the camp, however his sense of belonging towards the camp and people involved in the camp was not as high, also he did not find relevance of the camp activities to his life or as part of a career interest or hobby. Therefore, the activities in the camp caught his interest/attention, but his interest was not sustained over time.

Karla:

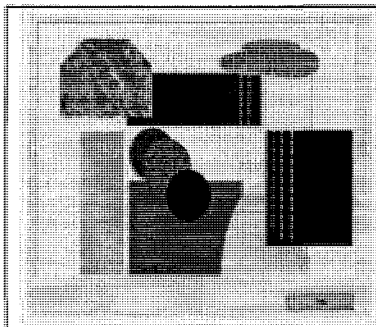
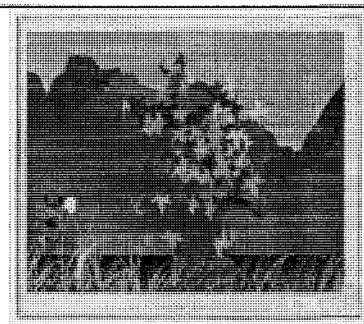
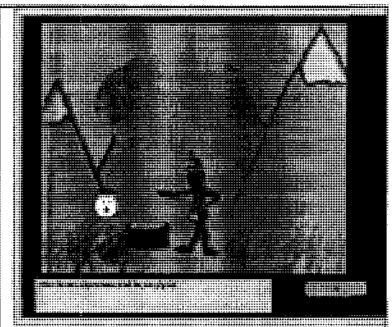
		
Programming Task #1	Programming Task # 2	Programming Task # 3
#Lines Code: 81	#Lines Code: 78	#Lines Code: 135

Figure 19: Karla's Programming Tasks

Karla was an 11th grader, and she represented the target population the summer camp aimed to attract (female with little prior computer experience). She did not enjoy doing mathematics and had limited knowledge about using computer technologies, but was attracted to the camp because of the animation and storytelling features advertised by it. Figure 19 show different screenshots of Karla's programming tasks.

Her previous knowledge with computers involved a keyboarding class she took in junior high, but she did not have any experience creating any type of animation or game.

In the initial survey about positive attitudes towards math, drawing, computers, games and animations, she indicated that she did not enjoy doing math at all, was neutral about working with computers, but enjoyed drawing, watching animations and playing computer and video games a lot; her favorite game was "Spyro: A hero's tail".

In terms of her interest for joining the camp she mentioned she wanted to learn how to create animations; however she was somewhat interested in creating animations, and did not have interest in learning to create games.

Karla's perceived self-efficacy for doing math was extremely low, and her knowledge about working with computers was also limited. Moreover, she perceived her skills for drawing as being average. Her low perceived abilities towards math and computers were confirmed when she was trying to accomplish the first programming task.

In the first programming task, Karla struggled a lot with the mathematics involved in understanding the coordinate system used by the computer to display graphics on the screen, which represent the lower points for the y-coordinate at the top of the screen and the higher points at the bottom of the screen so the coordinate (0,0) was located at the upper left corner of the screen. This student could not create a house or a scene composition; she was able to display geometric shapes and applied some texture brushes but had very little control on the composition. She also had some difficulty understanding the order of instructing the computer to place objects on the screen, so her work showed little grasp of purposely made composition.

Karla was able to finish the second programming task more successfully than her previous task, in part because the TAs were more aware of the problems she had with the programming and she received more help, also it was easier to follow the steps for this task by looking at the instructors' code displayed on the screen. She was able to get her background image to scroll and added a character created by the art instructor as an example of an animated character into her program. She did not integrate any original character into her program but used the background image she created. She did not integrate any music, storyline or timers in her code.

For the final project, she worked together with another girl that she became friend with during the camp and most of the final project was done by her partner. They integrated her background into the story and a few sound effects.

In the survey conducted at the end of the summer camp, Karla mentioned that she felt just a little bit comfortable with programming and she felt as if she was doing poorly at programming. Although programming was a little interesting for her, she was not excited about programming, had some trouble understanding it and found programming very challenging.

The activity she enjoyed the most in the camp was creating the clay character, even though she was not able to integrate her character into her final project. She also found learning Photoshop useful because she could edit pictures and post them on music websites in the future.

She did not express strong interest for creating games or animations in the initial survey and her interest towards creating games and animation remained the same at the end of the summer camp. She did not answer the follow up survey.

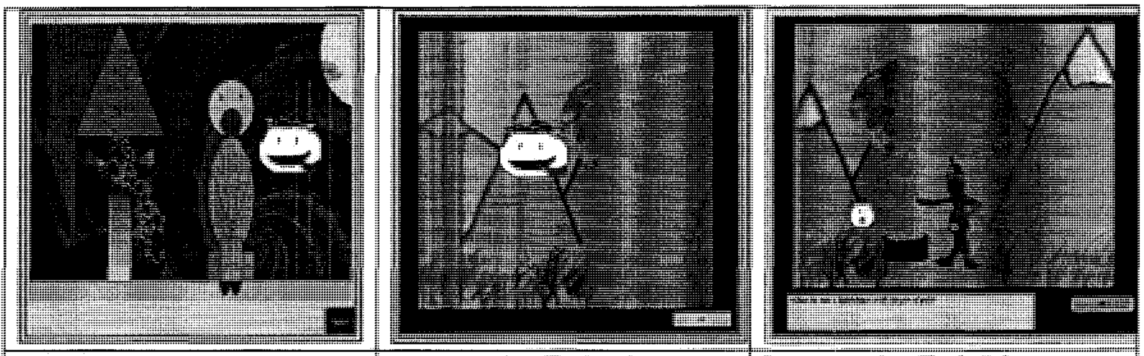
Karla's sense of relatedness was somewhat strong; she enjoyed the company of the rest of the students in the camp, made new friends and was planning on keeping in touch with some of them.

In terms of career interest, the things Karla did in the camp were somewhat in line with what she wants to do in the future as a career and fitted a lot with what she liked to do as a hobby. However she did not plan on continuing to work on her final project after the camp ended.

This student had difficulties understanding mathematics and low self-efficacy in her math and computer skill. As a result she had trouble completing the programming tasks, which were very mathematically oriented, and had low self –efficacy about programming, also she didn't find programming exciting.

Her sense of relatedness was relatively high, but her self-efficacy was very low, sense of control over the programming tasks was low, knowledge was limited, and she found the main activity in the camp (programming) very challenging. So it was very unlikely that her interest for creating interactive software was going to grow after this experience.

<u>Mary:</u>



Programming Task #1	Programming Task # 2	Programming Task # 3
#Lines Code: 97	#Lines Code: 84	#Lines Code: 135

Figure 20: Mary's Programming Tasks

This student was an 11th grader who had attended a technology summer camp called Digigirlz previously, but had not taken any technology class before. Figure 20 show different screenshots of Mary's programming tasks.

Regarding positive attitudes, Mary mentioned that she did not enjoy working with computers, but she enjoyed working on math problems, watching animations, and playing video games a lot. Her favorite games included all the "Final Fantasy" series and "The Sims".

She was interested in joining the camp in order to learn about animations, games, programming, and computers. She was somewhat interested in creating animations and in creating games.

Mary's self-efficacy in relation to math was very positive, but her knowledge about using computers was very limited. She considered her skills for drawing average.

For the first programming assignment, this student created a beautiful and complex composition. She understood the computer coordinate system very well; she created a house with triangular windows and placed them properly in the environment. She created a character by placing geometric shapes strategically to create the body and face of the character. Mary also demonstrated having a very good understanding of the order in which she needed to place objects on the screen. Mary also used the texture brushes very artistically, and there was a good sense of purpose in the composition, i.e., the sun was in the corner, clouds were in front of the sun, and rain stopped on the ground.

Mary finished the second programming activity successfully; she drew a simple character using MS paint and Photoshop and integrated into her program. She added sound effects, a storyline, and programmed two timers to change the storyline and the music at certain points of the story.

For the final project, Mary worked with Karla but Mary did most of the programming because she decided to improve her task 2 to create their final project. Mary spent a lot of time creating her clay characters; she created two clay characters for her animation and one prop. They incorporated Karla's background into the animation and programmed several events to support the story, such as making a character appear or disappear.

At the end of the camp when she was asked about her attitudes towards programming and the camp activities, Mary mentioned that she felt somewhat comfortable with the programming, and perceived herself as doing excellently in the programming activities. Although, she felt programming was very challenging, she thought it was easy for her to learn programming. Finally she mentioned she did not find programming interesting.

The activity she liked most was making the clay figure and putting it in the movie; she also really enjoyed incorporating the pictures and sounds in the program.

She had a strong sense of relatedness and enjoyed the company of the rest of the camp members a lot. She also planned contacting some of the students from the summer camp in the future. However, in the follow up survey she mentioned she had not contacted any of the summer camp members and had not received any encouragement from her school teachers to continue learning about technology.

As for career interest, she did not know what career path she wanted to follow and so the things she did in the camp were neutral for her career expectations and hobbies. So she did not find too much relevance in terms of career interest in this camp, therefore her interest for creating games and animation did not change throughout her experience in the camp. Even though in the follow up survey her interest for creating games and animations was the same as at the beginning, she was somewhat interested all the time.

After the summer camp ended, she mentioned that she might continue working in her project but in a follow up survey she stated she did not continue working on the project or creating any animations.

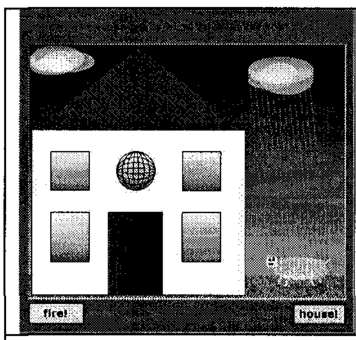
In summary, this student had a very good understanding of math and was very academically oriented, that combination helped her to succeed with the programming tasks. She knew very little about computers so she encountered some challenge when programming and using Photoshop. Mary's interest did not evolve over time as a result of this experience and she did not have any encouragement in her school environment to learn about technology so it was unlikely that her interest could evolve further.

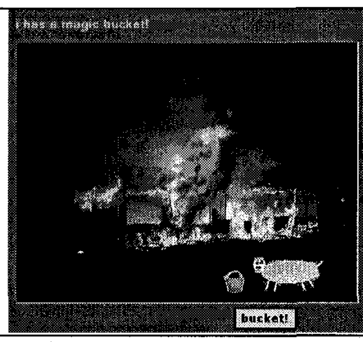
As a group Karla and Mary made an interesting couple: Mary was very academically and goal oriented. Whatever she needed to get done, she created a plan for that; she was very practical, and showed this through her research presentation and final project. Mary influenced Karla to keep working on the programming task but she didn't know how to make Karla cooperate with her so she did everything by herself.

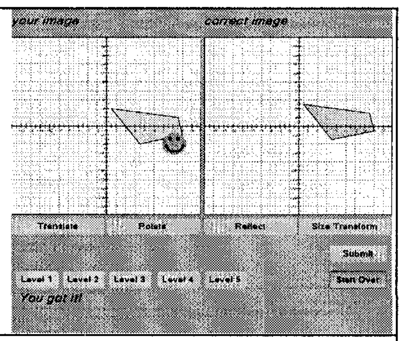
Teachers' Work

The first teacher chosen for case study analysis was Mark. He worked independently throughout the entire workshop; he did accomplish all the programming tasks very easily and had some previous knowledge about programming.

<u>Mark:</u>







Task #1	Task # 2	Task # 3: Final Project
#Lines Code: 88	#Lines Code: 37	#Lines Code: 661

Figure 21: Mark's Programming Tasks

Mark was a 9-12th grade math teacher, interested in creating games to teach mathematics. He had never created a computer animation but had created simple text based computer games. He was familiar with the process of creating computer games, and had taken programming classes in college. This teacher not only had previous knowledge about how to do programming but also was very familiar with the process of creating interactive software. Figure 21 show three screenshots of Mark's programming tasks.

Mark loved to teach math and working with computer technology, also he was an avid gamer and mentioned he had many favorite games to name them in the survey. As for his enjoyment of animations, he was fairly neutral about animations and did not enjoy drawing.

The main interest for Mark in joining the summer institute was the possibility of creating software that can teach his students a useful principle, in addition to the $700 compensation provided to the teachers in the workshop, which was very attractive for him.

Mark's self-efficacy for teaching math and his knowledge about using computers were very strong; however he thought he did very poorly when it came to drawing.

The programming task # 1 was very easy to do for Mark. He created a perfect scene, demonstrated an excellent understanding of the computer coordinate system and placement of the objects on the screen; he did understand the correct order for displaying objects in different areas of the screen through the programming.

Because programming task # 2 was easier and shorter than task # 1, Mark did not have any trouble accomplishing this task. He programmed an animation of a paint bucket that flew from the corner of the screen to the opposite end to reach a goat character.

This teacher worked individually on his final project; he created an educational drill and practice application to reinforce concepts of transformational geometry. He added sound effects from the video game Super Mario Bros and animated happy and sad faces that bounced all around when the user got a right or wrong answer. He spent lots of hours and days working on this project.

After the summer institute ended this teacher showed strong positive feelings towards programming; he felt very comfortable with it, and thought that programming was very interesting. His self-efficacy for how well he did in the programming tasks was very high, he also considered that learning programming was easy and was not challenging at all.

When Mark was asked about his favorite activity in the institute he mentioned that getting a chance to learn about graphics programming was very helpful as was creating a final project he could use in his classroom. Moreover, Mark found that what he did in the workshop was directly related to what he taught in school, so he planned on using the concepts in his final project to teach a lesson or two.

Mark's interest for creating animations decreased over time, as well as his interest for creating computer games to teach math and science. He was very interested in creating computer games at the beginning of the institute but at the end of the institute and two months after the institute ended his interest faded from very interested to somewhat interested in creating games.

Although, Mark's interest in creating computer games faded over time he was able to find relevance in the tasks he was doing in the summer institute and kept working on his final project after the institute was over, moreover, he was planning on having the kids play the game he made because it directly related to the unit they were studying, as he expressed in the follow up survey.

In terms of relatedness, Mark was a loner, he kept the interaction with his peers to minimum during the institute and did not plan on contacting anyone after the institute ended and he confirmed in the follow up survey that he did not contact anyone from the institute.

Rose:

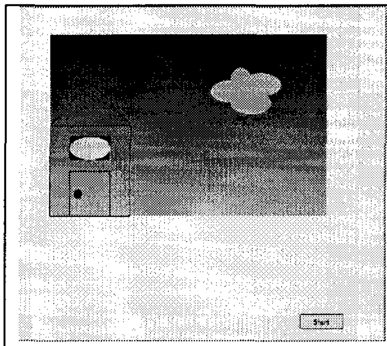
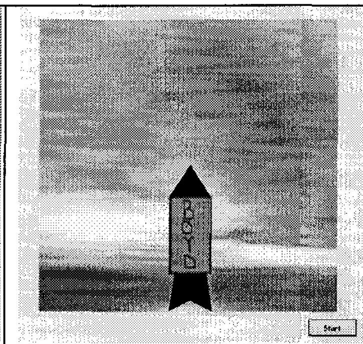
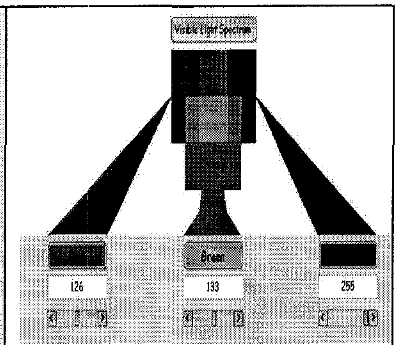
		
Task #1	Task # 2	Task # 3: Final Project
#Lines Code: 88	#Lines Code: 37	#Lines Code: 661

Figure 22: Rose's Programming Tasks

Rose was a 9th grade physical science teacher, she joined the summer institute with one of her coworkers and they were planning on working together for the final project. Figure 22 show three screenshots of Rose's programming tasks.

This teacher did not have any previous knowledge of how to create games or animations, however she had attended several classes to learn about productivity software such as Microsoft Office suite and file management.

Positive attitudes towards animations and games were low for this teacher she did not enjoy playing computer or video games at all and she somewhat liked to watch animations. Conversely, she enjoyed teaching math and science, working with computers and drawing.

Rose was somewhat interested in creating animations and somewhat interested in creating games. She was motivated to join the summer institute because she wanted to learn how to use animation to teach science and learn how to better use technology in her science class. Additionally, the five professional credits, $700 stipend, and the location of the institute (Seattle) were attractive to her.

She considered her self-efficacy for using computers, teaching math and science and drawing pretty much average.

For the first programming task she demonstrated a good understanding of the coordinate system and was able to place objects on the scene in the right order. For example, in the scene the sun was behind the clouds. Then for the second programming task she was able to display a rocket image in front of a background image and animate the rocket to move up as if it was taking off the ground. The last programming task, the final project, she paired with her coworker who was a technology teacher and he did most of the programming and she did the lesson plan for the final project.

Positive feelings and self-efficacy towards programming were very low for this teacher, she did not find herself comfortable doing programming, she found programming extremely challenging and difficult to understand, she felt as she did very poorly in the programming tasks, and she found it somewhat interesting.

Rose did not have a good sense of self-efficacy during the workshop however the experience in the summer institute became relevant for her because she learned how to integrate technology into her classroom, even more that what she already knew, and she felt that the animation and lesson plan she and her partner created as final project could be used in her classroom. She also gained a lot of new ideas for teaching in her class. As her favorite activity, she mentioned that she loved making her clay figure and working with her partner to create a computer animated simulation for her classroom.

Rose's interest for creating animations and games to teach math and science was middle ground from the beginning and it did not change over time. She kept selecting in the surveys the "somehow interested" options; moreover she did not continue working on her final project or any new animations. However, in the follow up survey she mentioned that she was planning using the animation that she and her partner made and about 6 animations that other teachers made in her Science class. Rose found relevance but her low self efficacy for programming did not allow her interest to create animations and games grow.

Patrick:

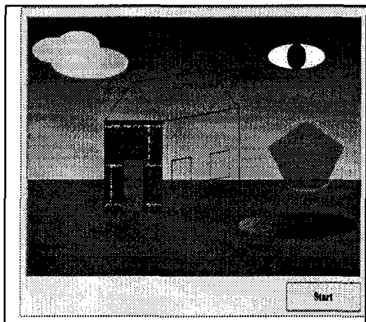
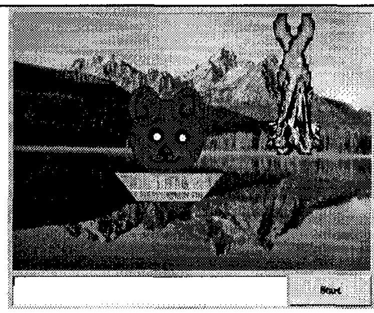
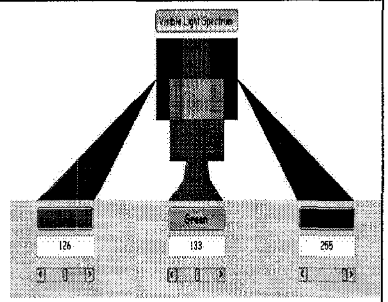
		
Task #1	Task # 2	Task # 3: Final Project
#Lines Code: 80	#Lines Code: 64	#Lines Code: 661

Figure 23: Patrick's Programming Tasks

Patrick was a 9th grader computer applications and robotics teacher who was very fluent in using computer technology. He had created simple animations before, by using PaintShop Pro and had created a number of math drill games on the computer. Figure 23 show three screenshots of Patrick's programming tasks.

This teacher did not find any enjoyment on watching animations, playing games or drawing, however he enjoyed working with computers and teaching about computer technology. His primary interested for joining the summer institute was to learn Visual Basic.

Ratings of interest in learning to develop games and animations were high for this teacher because he wanted to teach his students to use Visual Basic to create animations. He had a strong sense of self-efficacy about using computers and teaching mathematics and technology.

For the first programming task he created an interesting composition where he displayed a house on the screen with some perspective and created an eye using different ellipses. This demonstrated his good understanding of the coordinate system and programming logic to place objects on the screen.

In his second programming task he created a boat that moved from left to right and added a character he created into the scene.

Finally for the final project, Patrick and her partner worked together on a piece of educational software aimed at helping students understand how color is determined both physically and through computers (pixels). They created an RGB tutorial that showed the color obtained by combining specific amounts of red, green, and blue. Patrick incorporated his clay animation as part of the introduction screen, added sound to the application and provided an information screen with the light spectrum for the colors.

Patrick's sense of self-efficacy and positive feelings about programming were strong, he felt very comfortable with the programming, felt that he was doing well in the programming task, and that programming was easy to learn. Even though Patrick did very well on the programming tasks, he thought programming was a challenging activity. A possible explanation is that he created a complex final project and he faced several programming challenges trying to implement the necessary programming in his code.

The favorite activity for this teacher was the programming; however he appreciated having many knowledgeable people around that could answer his questions about programming.

Throughout the teacher institute, his interest for creating games and animations was high, in the follow up survey he rated the interest one point lower than before but it was still high overall. This teacher also kept working on his final project after the institute ended but had not created any new educational applications when the follow up survey was conducted.

Patrick was also planning using his final project in the future with his students and had been exploring a number of animation programming languages like *StarLogo TNG*, *Phrogram*, and *Scratch*.

Sense of relatedness for this teacher was also very high. He enjoyed the company of the rest of the members of the group and liked to have knowledgeable people around to share ideas and to ask questions related to the programming. The content of the institute was very relevant for him and he was already interested in everything the institute offered so his interest kept high and his knowledge about creating games and animations increased.

Final Projects

Students and teachers followed different work cycles for creating the final project. One difference between the groups was the amount of time given to the teachers to work in their final projects, one week, compared to a few hours given to the students to work on the final project.

The amount of instruction received by the two groups was also different despite the fact that the same instruction, content and material was provided to both groups. The instruction for the students was dense; students were given new content every day until the last day of the camp, when they had to do their final presentation. There was a lot of emphasis on giving them the entire content, which included extra material about how to use physics in their games and more sophisticated mathematical equations. Students also took a mathematical test from the camp instructor the last day that included questions related to physics and mathematics. This might be one of the reasons why there was so much emphasis on providing this new material at the end of the camp. The instructor used the test as a measure of how well students understood the mathematical and physics content he taught in the camp.

Teachers on the other hand were not as passive as the students about receiving new material every day; they complained to the instructors if he gave too much content and asked for more time to work on their projects. The instructor already warned the participants of the summer institute that he usually covered only a small fraction of his lessons with the teachers whereas he gave all the lessons to the kids.

The teachers received a teacher's programming guide and a student's programming guide, which differed in length (the students' guide being thicker than the teachers' guide). One of the teachers noticed that the teachers guide was much thinner than the student's guide and he asked the TAs as a joke if there was some type of implicit message about the abilities of the teachers and expectations about what they could accomplish.

The instructor spent the first week of class providing direct instruction to the teachers and the second week was devoted almost entirely to the final project.

The expectations were also different for students and teachers: students were expected to create a final project built upon small modifications of the programming task #2 so they did not need to start from scratch even though some students did, they did not need extra material or content to finish their project, they were provided with all the basic instructions and skills they needed to accomplish their final project. Teachers, on the other hand, were expected to create a piece of educational software that taught a principle or concept of the subject matter they teach at school; however teachers were not provided with the knowledge, tools, or minimal skills they needed to create the final project. Teachers had to buy Visual Basic books, and relied on the instructor and TAs help to create their projects, they did not know the type of skills or effort required to create some of the projects they envisioned, in some cases teachers expressed that most of their final project was created by the TA.

The amount of effort observed in teachers' final projects compared with students' final projects was also reflected in the number of lines of code teachers used in creating their projects compared to the students. Also, increments in the degree of difficulty to accomplish each of the programming tasks was better distributed and balanced for students, where their first task was somehow easy and did not require many lines of code, the second task was more complicated and required more lines of code and the final project required a few more lines of code than task #2. Teachers had to program more lines of code for the first task than for the second task, and the number of lines of code needed to create their final project increased exponentially compared with what they had developed before.

Students' Final Projects

The types of final projects created by students fitted into one of three categories: animations, games, and interactive animations. Animations did not include any type of user input, had a well-defined story line, background music and sound effect, and also had a well-defined beginning and an end. Games were highly interactive applications, have some sort of instruction about how to play it or the goal of the game, provided scores, visual feedback from users' input, and ended when the user got killed, or did not overcome an obstacle. Interactive animations were a combination of animation with user input, the user could control a character by making him jump, fly or walk and there was a sort of story express by the animation or text; also interactive animations had some background music and sound effects.

Students who created a game for the final project used more complex mathematical concepts and equations than the ones that decided to do pure animation; students who created interactive animations also used some mathematical equations to program their interactions.

In total there were nine final projects three were done individually and six were done in pairs. Two projects were animations, three projects were games, and four projects were interactive animations. An example of each type of project is presented in the following sections.

Animation:

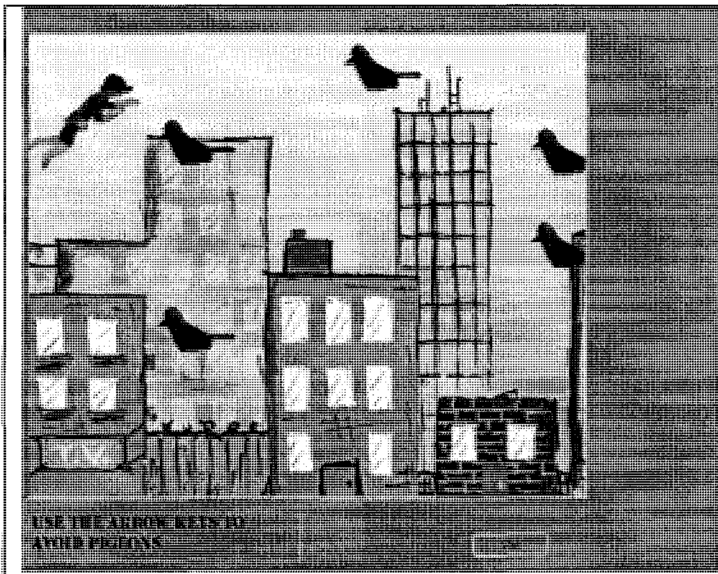
Animation #1: Carl and Cindy Lou, the love story

#Lines Code: 186

Figure 24: Students' Final Project (Animation)

Animation # 1, showed in Figure 24, was created by two female students who were attending the same school and were good friends. Both students were very talented artists, one student wanted to be a writer and the other one wanted to pursue a career in marketing. They worked together from the very beginning of the camp drawing the characters and background they wanted to use in their story using Photoshop. They cooperated very well to create the final project. Their animation was a love story between two parrots. They created very interesting zoom effects for the beginning of the animation to create the effect that the pirate ship was closer to the camera and then focused on the parrots that were standing on the mast of the ship. They used matrix transformations to accomplish this effect.

Game:



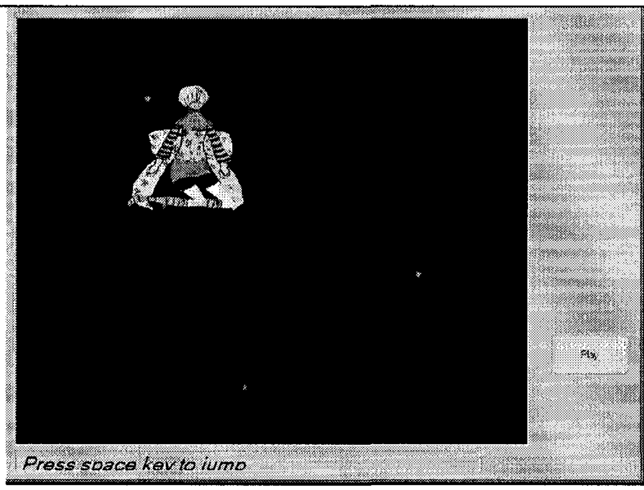
Game #1: Superdude

#Lines Code: 198

Figure 25: Student's Final Project (Game)

Game #1, showed in Figure 25, was created by a male student who was being home schooled, he created the final project by himself and mentioned in the survey that he got some help from friends. He wanted to become a graphic designer and was very knowledgeable about how games were created. He had done some claymation before and was very interested in getting more knowledge about the techniques to incorporate clay characters in the game. He spent numerous hours on his clay characters and incorporating them in his final project. His game consisted of a superhero flying around the city and avoiding being hit by pigeons. He used collision detection to make his program give some feedback when the superhero touched a pigeon.

Interactive Animation:



Interactive Animation: Ninja jumping
#Lines Code: 110

Figure 26: Students' Final Project (Interactive Animation)

This interactive animation, showed in Figure 26, was created by two female students, they did not know each other before the camp, and they were encouraged by one TA to work together in their final project. They collaborated to some extent, however they worked in their individual projects and presented to the class as two separate projects developed individually, they had a hard time integrating the previous art work and previous programming into the final project. The interactive animation shown above was done by the student participating in the present research; the other student was not part of the research. This animation consisted of a character that can walk back and forth and jump by responding to user input. There was some background music and one line of instruction to let the user know how to get the ninja to jump. The student brought to class a ninja character she drew in paper and scanned it and modified it in Photoshop to make the effect of the character jumping when the user pressed a key. She used the equation of the parabola to implement the jump in her program.

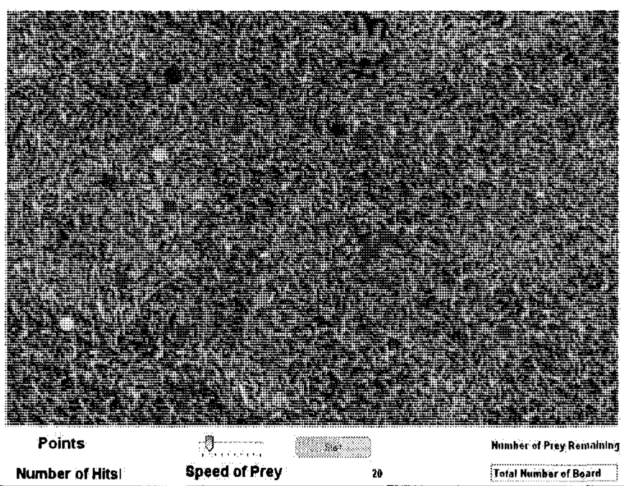
Teachers' Final Projects

Teachers' final projects represented three different types of educational software: simulations, drill and practice, and tutorials. Simulations were characterized by presenting a phenomenon or principle and by allowing the user to change parameters in the environment and see the results. Drill and practice presented a principle or an opportunity to practice a concept by doing an activity that resembles a game. Tutorials presented information about a principle or lesson and provided instructions to the participant about how to explore this concept or principle.

All the educational software needed to include as part of the final project requirements: one character that moves, one animated character, sound, and keyboard or mouse control.

In total, there were fourteen final projects: there were five done individually, and nine were done in pairs. Eight projects were simulations, three projects were tutorials, and three projects were drill and practice games. An example of each type of project is presented in the following sections.

Simulation:



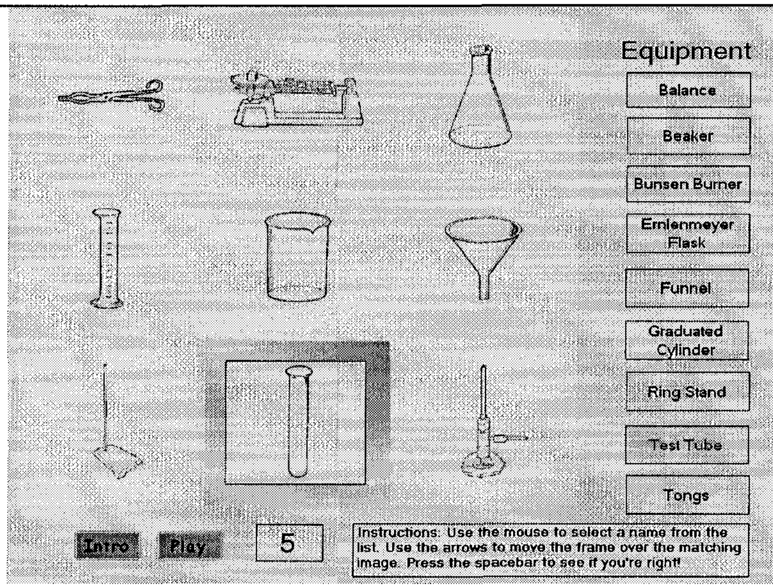
Simulation: Predator sprite

#Lines Code: 216

Figure 27. Teachers' Final Project (Simulation)

The simulation showed in Figure 27 was created by a female life science and programming teacher and a male biology teacher. They came from different schools but were seated one beside the other and decided to join forces to try to accomplish the final project. The female teacher had more programming experience than the male teacher, they both had several problems with the computers they were working on, and they lost their code several times, which make the programming experience very frustrating. They needed a lot of help from TAs and intructor to create their final project because they did not know the amount of effort they needed to program the project they envisioned. They indicated that they wished they knew that what they wanted to do was beyond their abilities before embarking in the project; they even lost the code for their final project when they were almost done and had to request the help of the TA to do it again. Part of their problems were based on: the instability of their computers, Visual Basic version running on their computers, and lack of understanding of the file structure created by Visual Basic. Their simulation portrayed the relationship between adaptation and predator/prey to drive evolutionary changes.

Drill and Practice:

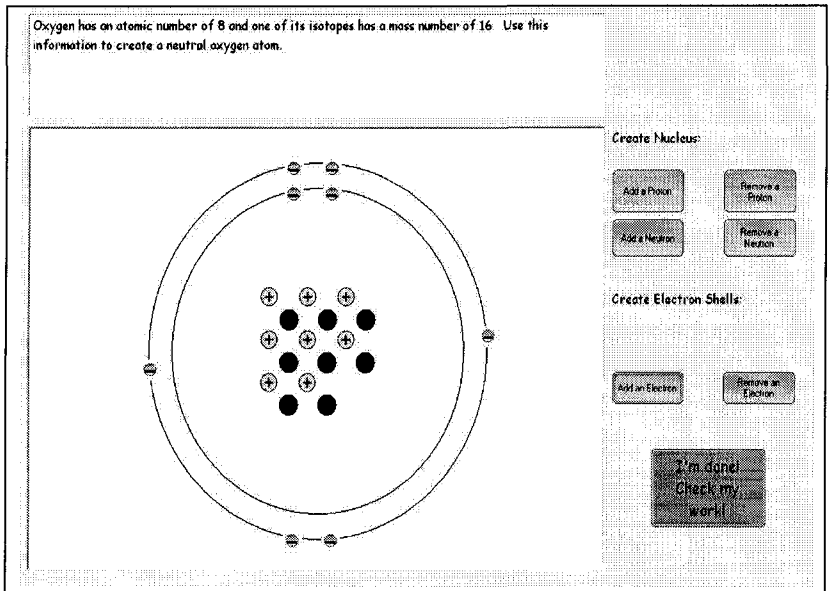


Drill and Practice: Chemistry Lab Equipment Matching Game
#Lines Code: 495

Figure 28. Teacher's Final Project (Drill and Practice)

The drill and practice chemistry game (see Figure 28) was created by an earth science and chemistry teacher who worked individually in his final project. This teacher mentioned that he made a bad decision by working alone in the final project. After he realized of the challenge that this represented, he got at least one Visual Basic book. In the post workshop survey he mentioned that the book did not help him too much in his final project and he relied a lot on the instructor and TA's help. His chemistry lab game was aimed to help students practice and test themselves on the names of some of the equipment they would use in the chemistry lab. He also received great feedback from the other teachers who mentioned they would like to copy the game and use it in their classrooms as well.

Tutorial:



Drill and Practice: Physics Tutorial
#Lines Code: 280

Figure 29: Teachers' Final Project (Tutorial)

Figure 29 show a tutorial created by a female biology and chemistry teacher and a physics, astronomy and chemistry female teacher. They were coming from different states and did not know each other before the camp. One of the teachers achieved better understanding of the programming than the other, and did most of the programming. One of these teachers improved a lot in her programming skills and comfort level with the technology which was very low at the beginning of the institute, however, she did not have enough programming knowledge to create the type of tutorial she wanted but she got helped from the instructor, TAs, and family members. The idea of the tutorial was to guide the user to build a Bohr model for a neutral atom, an ion and an isotope of a representative element. If the user did not accomplish the task correctly, some help and guidance was provided from a help screen.

Class Website and Online Community

There were two different class websites with similar content, one for the students group and one for the teacher group. Similarly, two independent online communities were created using Google Groups. See appendix D for example of websites and online communities.

Students frequency of use for the class website included “using it occasionally”, “rarely”, or “never”. The ones that used it mentioned they went to the website to review the video tutorials, access images and sounds, and to see other students’ work.

Only one student accessed the Google group; she posted information about favorite animations and questions but the only interaction in this online community was between the researcher and the student.

Teachers' response to the class website was more positive than the students' responses; they mentioned they used the class website "frequently", "occasionally", "rarelly" and "never". The most helpful features of the websites according to them were watching the TA programs, links to free images, programming hints, video tutorials, having many resources available, providing inspiration and motivation to get a good work posted on the gallery website.

The teachers' Google Group got 10 teacher members; however, there was minimal interaction among them. They prefered to email the researcher, instructor and coworkers for questions rather than posting questions on the online community.

Chapter 6:

Discussion, Conclusions, Recommendations and Limitations

This research was studied how high school students' and teachers' interest in creating computer animations and games changed when attending a summer workshop and during a period of two months after the workshop finished.

Because of underrepresentation of female and minority students in the science, technology, engineering, and mathematic fields, the summer camp tried to attract these types of students. Female students were successfully attracted to the camp but minority students were not. On the other hand, teachers enrolled in the summer institute were gender balanced and ethnically homogenous, but their skill levels were very diverse.

Five questions were formulated to study students' and teachers' interest, to compare the two groups and to understand the role of a website and online community. Quantitative and qualitative methods were used to analyze these questions. In the next section, a discussion of the findings is presented.

Discussion

Question 1: (a) How does students' interest in creating games and animations change during and after an animation summer camp? (b) How do students' previous experiences affect their interest and performance when creating interactive software? (c) What factors influenced students' interest for creating games and animations?

Analysis of the results for question 1 revealed that students' interest for creating computer animations was high at the beginning of the camp and did not change from the beginning of the summer camp thru the end of the camp. However, ratings of interest decreased two months after the camp was over but the difference was not significant. This result was not that surprising considering that students who participated in the camp were already interested in learning about animation and storytelling.

Students' interest for creating computer games was also strong but not as high as interest for creating animations. At the end of the summer camp, interest for creating games decreased, and two months after the summer camp was over it was still lower than when students started the summer camp.

Many students were interested in creating games at the beginning of the camp but by the end, their interest decreased. One possible explanation might be that most students did not know what it took to create a game in terms of programming before the camp started and, after learning how challenging it was to create games, their interest faded.

When students were asked about their future plans in continue working on their final project the majority showed their intentions to work on it or at least try to work on it. However, when they were asked in the follow-up survey if they indeed worked on their final project after the camp ended, only a small percentage of students indicated they did so, moreover a smaller percentage indicated they created a new animation or game. This indicated that their interest for what they learned in the camp was not sustained over time.

Correlating students' previous knowledge, performance on the programming tasks, and attitudes towards programming confirmed one of the findings several researchers had described over the years, which is that a good foundation of mathematics was very important to succeed in the programming tasks (Margolis & Fisher, 2002).

The degree of challenge was inversely proportional to how comfortable students felt about programming; the more comfortable they were with programming, the less challenge they felt towards doing programming. This last finding was slightly different that what flow theory suggests (Csikszentmihalyi, 1990). This motivation theory proposes that people feel the most rewarding experience when they have the right skills for the task, and the task is not too easy or too challenging for them. In the summer camp students were more comfortable when the challenge was low and their skills were high. However, changing the word “comfortable” in the survey for “rewarding” might generate different responses because sometimes rewarding learning experiences are not necessary easy or make students feel comfortable.

The correlation analysis also found that the stronger the knowledge students had about using computers, the less challenging they found the programming tasks. Also, performance on the programming task 2 (creating a side-scrolling and interactive program) was correlated to pre-knowledge about using computers.

There is still a debate about what knowledge related to using computers is useful to make the introduction to programming less challenging. The student survey did not ask for details in this regard; however it did ask about the type of classes students had taken in their school or on their own time. These classes included computer fluency, web development, typing, technology camps, and 3D modeling software.

However, by observing student participants and listening to their conversations during the camp, it was easy to tell that they had several informal learning experiences that helped them to develop their knowledge on using computer; those included moderating web sites devoted to games, playing video games, and interacting with family members who work in software development companies or who are professional artists. Also several students were very familiar with image editing software, such as Photoshop, Paintshop Pro, and PhotoImpact, which they had at home and use quite often.

Several research studies had pointed out the increments in home use of computers in the United States (Cohoon & Aspray, 2006) and the diversity of computer technology activities young adults engaged in informal environments, such as gaming, blogging, and instant messaging (Lenhardt & Madden, 2005). Students in the summer camp showed with a variety of background knowledge about how to use computer technology.

One of the factors that influenced students' interest for creating interactive technology positively was being able to see how the clay characters and digital drawings they created using Photoshop came to life when they integrated them into the program they created. The majority of students mentioned that the activities they enjoyed the most were working on their final project, seeing how the art, sound and programming fit together in a game, in addition to learning the craft of using Photoshop and claymation to produce animations.

Students who struggle with programming had trouble integrating their characters into their programs to create the animation, so they basically worked with another student to create their final project but were not able to contribute too much to the final project. This problem was also found in a technology program for girls called Techbridge (Kekelis, Ancheta, Heber, & Countryman, 2005).

Another factor that seemed to motivate students was looking at previous-year animations and professional animations in class. From observations and students' feedback, they were very engaged in looking at other people's animations and game creations. Although there was not too much opportunity for debriefing or reflecting about the animations with the current structure of the camp, this activity provided students with ideas about what other people with different level of skills (professional and students) had accomplished before and served as an inspiration for the kids. Moreover, Duesing and Hodgins (2004) found that showing short animations relevant to students' projects encouraged students to think more broadly about this art form.

The curriculum was very effective in the way it presented the use of mathematical concepts applied to computer graphic programming and was very helpful in connecting mathematical concepts to solve graphic problems such as scrolling a background image, or detecting a collision between two objects. All the students were able to implement mathematical concepts into their programming. Some of them required more help to accomplish this task but in the end, all of them got plenty information and examples of how different mathematical concepts could be used for programming computer games.

Finally, the flexibility of the program in allowing participants to create different types of applications with the knowledge they gained was very effective. Participants who did not like games could create animations or interactive animations, and participants who liked games could create mini games or something in-between games and animations. Flexibility to explore different domains of computer technology is one of the key elements to impact attitudes towards computers in a positive way (Jones & Clark, 1995).

Additionally, researchers have found that when students with different background knowledge about games and computer applications are asked to design games, the ones with more experience playing games (usually the boys) develop products that resemble what they usually play or contain features they identified with games, such as certain types of gameplay, feedback, interface, etc. When students who do not play games regularly (usually the girls) were asked to design games, the results were a little different than the ones designed by students who played games regularly. They created applications that do not typically behave like commercial games do, and provided different type of feedback and activities than the ones associated with existing games (Denner & Campe, 2006).

Providing the flexibility to build different applications allowed students with different knowledge to associate their previous knowledge with the type of application they wanted to create.

Some of the factors that did not increase students' interest for creating games and animations included using a professional development environment to do programming, such as Microsoft Visual Basic, without enough explanation of the tool.

Because participants were exposed to the same tools professional programmers use, they also faced the challenge that comes with using complex development environments, such as how to handle project folders and multiple files. Some participants were prepared to work with this type of professional tool but most were not prepared for that. This was especially true because the instruction about how the files are organized in Visual Basic and how the compiler works were not included in the workshops. TAs commonly fixed participants' code when they had any type of error, and because of that, students' sense of autonomy, i.e., independence to fix their programming code was limited. Students also had several problems trying to integrate previous versions of their codes or their partners' code into one program.

Using Visual Basic provided many advantages for teaching graphic programming, because it is simpler than most programming environments but it does not hide or oversimplify the mathematical concepts needed to do graphic programming. However, more instruction about the compiler and the file structure was needed.

Certain types of social interactions decreased students' interest for creating games and animations. One type of interaction was peer influence. An example came from observations of students interacting in the camp. One of them started the camp very motivated but after he made friends with another boy who showed very little interest in the activities of the camp (played games on the computer and cell phone during class, didn't finish the programming tasks, etc), his performance and engagement in the programming task diminished. Another example came from observations of a student who did not make any friends or socialize with the rest of the students. She started working on her project with some enthusiasm, but at the end she lost part of it, she didn't incorporate any storyline to the animation even though she was encouraged to, and didn't stay for the final presentation because she had another engagement and asked permission for the instructor to do her final presentation early in the day without the parent audience.

The most disconcerting peer interaction happened when an African American student brought a friend to the summer camp because she did not feel she had anyone to talk to in the camp. Her friend was accepted and helped her to keep on task for the first two days, but after that she became a distraction and did not help her summer camp friend to create a final project. Previous research suggest that a stereotype thread might have been perceived for this student in situations like this, and she might have tried to overcome that by bringing a friend to the camp. However, her friend distracted her from accomplishing some of the tasks in the camp and limited her ability to make new friends or collaborate with others. As a result, her performance, self-efficacy, and sense of relatedness were not the best. This type of friction in technology camps with ethnically diverse students have been also found and documented by Kekelis et al. (2005) and Goode (2008).

The focus on covering (quickly) all the mathematics and physics material related to graphic programming, and introducing new concepts without giving enough opportunity to the students to process the previous concepts did not help promote motivation, learning and studio thinking habits (Bransford & Brown, 1999; Hetland, et al., 2007), because there was no time for reflection about what was being learned.

Additionally, testing students' knowledge in mathematics with a traditional school test caused a disruption in the flow of the class, because students had to stop working on their projects the last day of class just a few hours before the final presentation to take a test. Evaluation in the form of written test might have created an additional stress in particular for students who do not like doing mathematics; also, evaluation decreases intrinsic motivation (Sansone & Morgan, 1992). Better ways to evaluate students' understanding could include asking them to reflect on what they learn about how mathematical concepts such as matrix translations are applied to computer graphics in a small group discussion, and then have them present it to the class.

Question 2: (a) How does teachers' interest in creating games and animations change during and after an animation summer institute? (b) How do teachers' previous experiences affect their interest and performance when creating interactive software? (c) What factors influenced teachers' interest for creating games and animations?

Teachers' interest in creating animations and game technology was strong at the beginning of the summer institute, and increased at the end of the summer institute, and then decreased two months after the institute was over. From the statistical analysis, only interest for creating animations decreased significantly between the time the summer institute ended and two months after it ended. Overall, teachers had a positive experience at the end of the summer institute; they expressed happiness and gratitude about what they learned and were grateful to take home their educational software and some of their classmate educational software as well.

At the end of the summer institute, most teachers had the intention to continue working on their animation and only 5.26% said they did not plan on continue working on it. However, the follow up survey revealed that only 31.25% did actually work some on their final project after the institute was over, and 18.75% had created a new animation or game.

A possible explanation for teachers' increased interest in creating interactive software between the beginning of the institute and the end could be the positive experience these participants had towards the end of the institute, specifically the last day. At that time, they made their final presentation, were rewarded with positive feedback from their peers, and were given several gifts from the instructor such as photo editing software, laptop bags, blank cds, etc. Teachers' final presentations were done in a friendly environment and they seemed to enjoy looking at their peers' final projects. They provided each other with positive feedback, and asked questions about other people's software functionality and requested to copy some of their software.

The reasons for the decrease of interest after the summer institute might be that most of the teachers were able to complete their final project only with substantial help from the instructor and TAs. Later, when they did not have support, it might have felt challenging to continue working on their final projects or creating new Visual Basic programs. Some of the teachers even started exploring different programming environments that suit their students better because they found Visual Basic too complex to use with their students. Supports from others played an important role in developing interest, without this support, development can become dormant, regress to a previous phase, or disappear altogether. Additionally, learners with emerging individual interest may need encouragement from others to persevere when confronted with difficulty (Hidi & Renniger, 2006) which was absent for most of the teachers once they went back to their schools.

Several relationships were found among teachers' attitudes towards programming, performance on programming task 1, and previous knowledge about using computers. For example, teachers that rated programming as interesting felt comfortable about programming computers. Similarly to the students, teachers did not felt comfortable doing programming when they perceived programming to be challenging. Also participants with strong knowledge about using computers were more comfortable with the programming tasks, performed better on programming task #1 and thought programming was something easy to learn for them.

One of the main motivational factors teachers experience in the institute was the ability to create a final project related to the subject matter they were expert in teaching, and being able to take the educational software to their classroom as part of their teaching tools. One way to increase people's interest for an object or activity is to associate the activity or object with a topic about which participants had a well developed interest such as the subject matter they teach in school. This way, there was meaningfulness in what they were doing as a final project and the value for that project was increased because most participants were planning on using their software in the future in their classrooms.

Interest theory stated that meaningful and personally involving activities can contribute to the maintain situational interest (Hidi, & Renniger, 2006). These activities can help integrate new knowledge with previous knowledge. Such activities can also provide additional relevance and value for the activity (Renninger, 2000). For example, teachers were able to articulate what they wanted to accomplish in term of their lessons and use that knowledge to ask questions about how to create interaction and provide feedback using programming.

Several participants were attending the summer institute along with coworkers; this helped to release the tension of working with new people on the final project and increased their comfort level to be part of the institute. Most teachers were also very social and they got along well with the rest of the institute members so their sense of relatedness was high.

One of the main factors that did not contribute to increase teachers' interest was the insufficient content presented to the teachers to build the skills necessary to create the final project. The content was very limited in comparison with what they were expected to do in their final project, also they were asked to implement keyboard and mouse interactions but they received very limited instruction about how to do it. Teachers' self-efficacy did not develop because they were not provided with the knowledge and skills they needed to successfully manage and develop their final project.

Similarly to what happened with students, teachers needed more instruction about how to use the compiler and the file structure created for visual basic projects. Without this knowledge teachers were powerless when it came to fix the errors their programs have or copy the right files in new projects. For this reason their sense of autonomy suffered during the summer institute.

Teachers and students had a hard time trying to collaborate doing programming. Particularly, teachers expressed their concerns about not knowing how to integrate their codes with their partners because of the complexity of Visual Basic project file structure.

Research Question 3: What are the differences and similarities between students' and teachers' levels of interest throughout the process of creating games and animations?

Students and teachers showed no significant changes in their interest for creating games and animation after attending the workshops. There was also no significant difference between the interest for creating interactive technology based on the type of participants (students or teachers). Perhaps, the length of the workshop was too short to increase participants' interest and the selection of participants may also have affected the outcome of the workshop because of the flaws found in the recruitment process.

Stress level was different for students and teachers at different points in time. Students showed less stress when they faced technological problems, such as saving the wrong files on their thumb drives or dealing with new information about using the computers. Teachers, on the other hand, showed a lot of stress when they could not save the right files on their thumb drives or were presented with new knowledge related to using the computer.

Once again the level of comfort young people had with technology was confirmed, it was higher than the comfort older people experienced. Students were very comfortable with the technology and it was easier for them to accomplish the first programming task than it was for the teachers. Interestingly enough, teachers rated their knowledge about computers stronger than did students. Yet, students performed better on task #1, were more creative about composing a scene with a house, and needed less help from the TAs than teachers did. Finally, students have more positive attitudes towards programming, have a stronger sense of competence, and perceive programming to be less challenging than teachers did.

The previous results are aligned with studies that had found that diversity of computer experience (but not quantity) was positive related to positive attitudes about computing (Jones & Clark, 1995). Students in general stated that they regularly engaged in diverse classes and activities involving computers, such as gaming, photo editing, and blogging, whereas teachers engaged in more traditional computer activities, such as emailing, searching the web, and using productivity software.

Some differences observed when comparing adult learning and children learning included: students did not question the information that was being presented to them whereas teachers frequently questioned what they were being taught. Students only listened to the teacher instruction and did what he said, while teachers took meticulous notes about what the instructor was saying; moreover students occasionally used commentaries in their code, whereas teachers' code had plenty commentaries about what sections of the code were supposed to do and tips about the programming.

Finally students were required to take several mathematical written tests which disrupted their learning activities and for which they did not get any feedback about the outcome, while teachers were not exposed to this type of test.

Research Question 4: What kind of games or animations did students and teachers create as a final project? How different was the process to create the final project for students and teachers?

Students' final project fit in one of three categories, animations, games, and interactive animations. The most popular genre for students was interactive animations, where they incorporated interactions they learned in class such as making a character jump, walk or fly by using the keyboard. Teachers developed either a simulation, drill and practice, or tutorial software; the most popular type of software was simulations, maybe because what they learned in the workshop was more appropriate for creating this type of software. Similarly, students learned more about how to create an interactive animation than how to create a game or animation.

Effort expenditure to create the final project was different for these two groups. Students were able to integrate what they learned into their final project in a matter of a few hours; however teachers had to work many more hours on their final project because they started an ambitious project from scratch, which can be very intimidating for novice programmers. Teachers needed more personalized help than students because of the complexity of their final projects; teachers also work at home or in their hotels, and bought books to be able to finish their projects. It is also important to mention that creating educational software is a very challenging task because developers need to be able to create interactive and engaging activities that promote learning opportunities, as has extensively pointed out by Alessi and Trollip (2001). Knowing the most effective way to teach a concept using the right format of educational software and the right feedback requires expertise in instructional design and software development.

Students did not have the opportunity to share their final projects with the rest of the class, while teachers were able to exchange software the last day of class. Exchanging the software was perceived as a very positive outcome from the camp because teachers found value not only in the product they created but in the product their classmates created, this was not the case for students.

Research Question 5: How did teachers and students use the class web site and online community?

The class website proved to be more useful for the teachers than for the students. Teachers used the website to access resources such as images, source code and links to get free sound effects, in addition to reviewing the video tutorials. Students used the website mostly to see the gallery with screenshots of students' work and animation videos. Perhaps the website was more useful for the teachers because they were more interested to learn from written material and video tutorials than students, additionally, teachers had more trouble understanding Visual Basic than students.

The online community was unsuccessful for both groups, it seemed like the community did not have a clear purpose and neither students nor teachers could find a need to use it. Also, to use the Google Group, participants needed to create an account and learn how to post information on the group. The researcher thought it was intuitive to use the Google Group but the results showed that it was not that intuitive. Finally, more facilitation was needed to build a physical community that could translate in an online community.

Conclusions

Participants came to the workshop with different levels of interest and knowledge for creating interactive software. Participants with less developed interest were expected to engage in the workshop activities, increase their knowledge for creating games and animations, and find relevance and value for creating this type of software. If participants accomplished the integration of knowledge, affect and value then interest was expected to grow at the end of the workshop. Additionally, if the experience in the workshops was meaningful enough to spark participants curiosity for developing more animations and getting deeper knowledge about creating games and animations then their interest would keep strong even two months after the workshops ended.

Other important factors for sustaining interest are embedded in the learning environment and activities: if the activities promote a sense of autonomy and competence in participants, it was more likely that interest would increase or remain strong. Also if participants learn in a supportive and friendly environment where they feel a connection with the people participating in the same learning experience, then participants' interest has more possibilities to grow.

The summer camp was successful in catching students' attention for creating games and animations by providing engaging activities such as creating digital art, integrating art work with sound and programming, and creating clay characters; however it failed in providing a safe and supportive environment for students with different backgrounds and skills, and in providing relevance and value for students attending the summer camp. As a result, students' interest for creating interactive technology did not increase after the summer camp was over, and faded two months after students participated in the summer camp. Relevance, support from others, and re-engagement in tasks that broaden students knowledge about creating games and animations played a crucial role in developing interest. There are enough evidence supporting the claim that engagement, support and relevance deepens interest for content (Hidi, & Renniger, 2006; Goode, Rachel, and Margolis, 2006)

The teacher institute, on the other hand, provided lots of meaning and value for teacher participants, especially when they were involved in creating the final project. However, it was unsuccessful in giving teachers the instruction and skills they needed to have more autonomy in the process of creating their projects. As a result, teachers' interest for creating interactive technology seemed slightly stronger than students, at the end of the teacher institute, but it faded two months after the institute was over.

If the workshops' success in motivating students and teachers to create interactive technology is measured by looking at what participants did in the final project, and by positive feedback from survey, the workshops might seem very successful. However, if we try to find increments in interest to create interactive software, it is difficult to find any significant changes in participants' interest. Even worse, I found a decline in interest for creating interactive software over time, which was not significant for the students, but was significant for teachers.

The results from this study indicate the importance of tracking participants' changes after getting involved in this type of learning experience aimed to boost interest and motivation for science and technology. More than 10 years of Tech summer camp experience have been collected across North America to increase female and minority student participation in the STEM fields, in particular computer science and technology oriented careers. However, few changes have been observed during the past 10 years in recruitment and retention of these students in STEM fields.

Previous research had shown the benefits of using project based learning, studio learning approaches, and cognitive apprenticeship to enhance motivation and learning (Collins, Brown, & Holum, 1991; Hetland, et al., 2007; Cash, Stadt, Behrmann, and Daniels, 1997). Some elements of these learning approaches were observed in the summer workshops but their implementation missed several important steps, such as the reflection on each person's work and their ability to provide meaningful feedback to others.

Finally, having clear criteria about the type of participants that would benefit most from this type of workshops, and making sure the type of participants that are accepted to join the workshops match the criteria, proved to be crucial for the success of the workshop. In the case of the summer camp, there was lack of diversity even though the camp was supposed to serve girls and underrepresented minorities. Also there were several highly talented students and students who came from the best private schools in the Seattle area with strong mathematics and computer knowledge backgrounds accepted in this summer camp that was supposed to serve primarily students who are not engaged with mathematics and technology.

Accepting students who do not represent the typical participant the camp aimed to recruit does create a problem. As happened here, it can be the case that the best students set the bar for achievement on the final project and that level may be too high for the rest of the average participants, and thus the sense of competence for participants with low performance and knowledge gets affected by the social comparison. Also, the instructor usually showed the best games and animations to outsiders and new workshop participants, and in the case of this particular summer camp the best software was developed by a student who came from one of the best private schools in the Seattle area.

Another good project was created by a student who was awarded with the National Center for Women and Information Technology (NCWIT) Award for Aspirations in Computing three months after the summer camp ended. This award was granted nationwide to 28 students who demonstrated outstanding aptitude and interest in: information technology/computing; solid leadership ability; good academic history; and plans for post-secondary education.

Teachers' recruitment was also deficient: the summer institute mistakenly accepted participants with professional programming backgrounds and nobody knew about this until later on. Participants in the institute noticed that some teachers were exceptionally good at programming but did not know it was because their strong programming background. This issue also revealed different goals and expectations teachers had in mind about the summer institute, for example, a group of teachers were interested in teaching their students how to create games and animations and had strong technology backgrounds, while another group of teachers only wanted to be able to use the software they created in their classrooms, and the last group was interested in the rewards provided by the institute such as professional development credits, cash, and a trip to Seattle. Lacking knowledge about participants' background, experiences and goals made the process of engaging in the content and helping participants with the tasks more challenging for the instructor and TAs.

Last, having beautiful and elaborated final projects was not necessarily compatible with increasing participants' interest to create interactive technology. Better results could be accomplished by letting students and teachers create some final product they can have some ownership and sense of pride, recruiting participants with similar skills and background levels, and providing choices about a broad range of projects participants can create where social comparisons are not going to damage less skillful students or teachers.

Recommendations

Some of the recommendations were collected as a reflection of the process of increasing students' and teachers' interest for creating games and animations and from other researchers' experiences promoting motivation.

First, simple techniques, -ones that do not require heavy programming- should be provided to participants, so they could put together the animations they created from clay figures and digital drawings using different techniques before moving to the programming. This way they could build more confidence and be able to see the final product of bringing their clay or drawings characters to life after spending a substantial amount of time working on them.

Secondly, helping participants, in particular students, to figure out what they are good at and enjoy doing when using technology would be a good outcome of the workshops. Letting them work on the things they are good at and feel comfortable doing can increase their motivation and create a more positive experience.

Third, even though, the workshops were about programming, participants might enjoy having the opportunity to explore other areas of the games and animation industry, such as the design process and software testing using standard methodologies used in the profession. Animation and storytelling techniques used by professionals in the field might be very engaging for most students also.

For creating diverse summer camps it would be convenient to get staff with experience in multicultural education for recruitment purposes and to facilitate interactions among ethnical diverse groups that are not used to work together (Kekelis, Ancheta, Heber, and Countryman, 2005). Also recruiting students who already know each other and can work well together would be beneficial (Goode, 2008).

Criticizing previous animation work is an important step to inform participants about what it takes to create an elaborated or simple final project and to encourage the development of studio thinking habits, such as envisioning the final project, and being aware of the process people go through to create their projects. Moreover learning techniques artist used to provide feedback and criticize art work would be very helpful for all participants. An example of how to teach to critique students work is provided in Whittington (2003).

Building trust and creating activities that promote collaboration and highlight similarities among the students, and encourage acceptance of their difference, is crucial to the success of programs like this, where students are encouraged to work together. Also to facilitate opportunities for students to collaborate or to work on their own pace would be beneficial for integrating diverse participants with different skill levels (Kekelis, Ancheta, Heber, and Countryman, 2005)

Revising the summer institute and summer camp's main goals seems very important. The summer institute was very successful in encouraging teachers who were not that knowledgeable about creating technology to use more technology in their classrooms and getting ideas about what interactive software other teachers created to present a concept or principle. However, the summer institute focused on teaching participants how to create interactive software instead of how to integrate it in the classroom, which creates confusion and stress for participants which were trying to build animations and games from scratch without having enough skills to do it. On the other hand, a group of teachers were there to learn how to teach their students to create games and animations and had the skills to do the programming but received very limited instruction about how to introduce novice programmers to Visual Basic and how to tell them about what happens behind the scenes.

The summer camp accomplished the goal of attracting female students to the camp but showed a conflict of interest when motivating students who came from different socio economic backgrounds and had been exposed to different opportunities to broader their knowledge about technology and study habits. Reinforcing the interest of students, in particular girls, who are already interest in technology, are good with technology, and had supportive environments at home to deepen their knowledge might contribute to help these students to consider careers in technology in the future if they already have not done it. On the other hand, bringing high performance students to this camp creates a social comparison that harms students who had not built the skills to perform that well in school environments. Rethinking the goals of the camp might help to define what the priorities of the camp are and take actions in that direction.

There was a need to explain to participants the competitive advantage of having technical skills if they want to do animation or want to take an art career path. Careers in computer graphics are very competitive and sometime the way animators survive and make their skills more marketable is by getting extra technical skills. This was not addressed in the camp, instead there was a presentation on the use of numerical models and animation in different professions and a presentation of salary range based on profession. Also, there was a discussion about the kind of mathematics courses students can take in high school if they want to go into computer science careers, which motivated several students to publically ask questions to the TAs presenting this.

Discussions on salaries and skills are very helpful but they required deeper discussions than just presenting facts.

Finally, a more honest approach, especially toward students, should be implemented. It did not seem fair from a motivational standpoint, for students who did not enjoy doing mathematics and had accumulated stress and low self-competence associated with doing mathematics, to join a camp that was advertised as an animation and art camp, when in reality they were doing mostly programming and learning new mathematics and physics concepts every day. Although there were some activities related to creating animations and creating a storyline, without doing programming students could not make their animations or tell their stories, which is not the case in real production environments where animators and storytellers do not necessarily had to be involved in the programming tasks.

Limitations and Future work

One main limitation of this study was that the researcher was the only person scoring the programming tasks, which is not the best approach to assess participants' performance. However, the researcher compared the ratings that participants gave to themselves about their performance and found similarities between the two scores.

Because of time constraints, the researcher did not notice the challenge it might have been for participants to access and participate in the Google Groups intended as a virtual community, for this reason more research in this area might bring new insights about how participants interact with each other when they have well-designed online communities to participate in.

Also, several teachers mentioned that they were using the educational software they created in the summer institute in their classrooms or were planning to use it, so it would be beneficial to see how teachers used their creations in their classrooms, and how their students would react to them.

Finally, the animation industry lacks women and minority workers, similarly to the information technology industry, for this reason, summer camps and summer institutes seem like wonderful opportunities to provide students and teachers with the techniques, skills, and knowledge real animators used in the industry to create their products, which require a different approach than a programming class. Increasing participation in computer graphic fields needs more than just teaching about programming and engineering; young people should be able to obtain information about broad range of careers in technology fields.

Bibliography

Ainley, M., Berndorf, D., and Hidi, S. (2002). Interest, learning, and the psychological processes that mediate their relationship. *Journal of Educational Psychology.* 94(3), 545-561.

Ainley, M., Corrigan, M., & Richardson, N. (2005). Students, tasks and emotions: Identifying the contribution of emotions to students' reading of popular culture and popular science texts. Learning and Instruction, 15(5), 433-447.

Ainley, M., Hillman, K., & Hidi, S. (2002). Gender and interest processes in response to literary texts: situational and individual interest. Learning and Instruction, 12(4), 411-428.

Alessi, S., and Trollip, S. (2001). Multimedia for learning. Methods and development. Pearson Ally & Bacon. 3rd Edition. Sydney.

Artemis project summer camp (2005). Retrieved June 10, 2005, from http://www.cs.brown.edu/people/orgs/artemis/home.html

Becker, K. (2001). Teaching with games: the Minesweeper and Asteroids experience Journal of Computing Sciences in Colleges, 17(2), 23 – 33.

Beyer, S., Rynes, K., Perrault, J., Hay, K., and Haller, S. (2003). Gender differences in computer science students. *Proceedings of SIGCSE'03,* Reno, Nevada.

Bransford, J.D., Brown, A. L., & Cocking, R. R. (Eds.). 2000. How People Learn: Brain, Mind, Experience, And School (expanded ed.). Washington, D.C.: National Academy Press.

Brown, J.S., Collins, A., and Duguid, P. (1989). Situated Cognition and the Culture of Learning. *Educational Researcher,* 18(l), 32-42.

Cash, J., Stadt, R., Behrmann, M., and Daniels, H. (1997). Effectiveness of cognitive apprenticeship instructional methods in college automotive technology classrooms. *Journal of Industrial Teacher Education*, 34(2), 29-49.

Cohoon, J.M, and Aspray, W. (2006). Women and Information Technology: Research on Underrepresentation. The MIT Press.

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience.* HarperCollins Publisher. New York.

Jones, T., and Clark, V.A. 1995. Diversity as a determinant of attitudes: A possible explanation of the apparent advantage of single-sex settings. Journal of Educational Computing Research 12(1), 51-54.

Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. *American Educator, 6*(11), 38-46.

Collins, A., Brown, J.S., and Newman, S.E. (1989). "Cognitive Apprenticeship: Teaching the Craft of Reading, Writing and Mathematics! In L.B. Resnick (eds.) *Knowing, Learning, and Instruction: Essa in Honor of Robert Glaser* Hillsdale, NJ: Erlbaum.

Cordova, D. I., and Lepper, M. R. (1996). Intrinsic motivation and the process of learning: beneficial effects of conceptualization, personalization, and choice. *Journal of Educational Psychology* 88(4), 715-730

Cox, D. (1991). Interdisciplinary collaboration case study in computer graphics education: "Venus & Milo". *Computer Graphics*, 25(3), 185-190.

Denner, J., and Campe, S (2006). What do girls want? What games made by girls can tell us In Y. Kafai, C. Heeter, J. Denner, and J. Sun (pp. 35-48). UCLA, Los Angeles: NSF Workshop.

Digigirlz (2008). Retrieved May 2, 2008, from https://www.microsoft.com/about/diversity/programs/camps.mspx

Duesing, J., and Hodgins, J. (2004). Team teaching animation art and technology. Paper presented at SIGGRAPH 2004.

Flanagan, M., Howe, D., and Nissenbaum, H. (2005). New Design Methods for Activist Gaming. Digital Games Research Association Conference.

Gamemaker. (2008). Retrieved May 2, 2008, from http://www.yoyogames.com/make
Girls, adventures in mathematics, engineering, and science summer camp (2005). Retrieved June 10, 2005, from http://www.wie.uiuc.edu/games/

Goode, J. (2008). Increasing Diversity in K-12 Computer Science: Strategies from the Field. *Proceedings of SIGCSE'08,* Portland, Oregon.

Goode,J., Rachel, E., and Margolis,J. (2006). Lost in translation: Gender and High School Computer Science, in Woman and information technology. Edited by J. McGrath Cohoon and William Aspray. 89-114.

Grolnick, W. S., Deci, E. L., & Ryan, R. M. (1997). Internalization within the family. In J, E. Grusec & L. Kuczynski (Eds.), *Parenting and children's internalization of values: A handbook of contemporary the*ory. New York: Wiley.

Guzdial, M. and Tew, A.E. (2006). Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. *Proceedings of ICER'06,* Canterbury, Uk.

Hardnett, C. (2008). Gaming for Middle School Students: Building Virtual Worlds. Paper presented at the Third Annual Microsoft Academic Days Conference on Game Development (GDCSE 08).

Hetland, L., Winner, E., Veenema, S., Sheridan, K., Perkins, D. (2007). Studio Thinking: The Real Benefits of Visual Arts Education. Teachers College Press.

Hidi, S., & Harackiewicz, (2000). An interest researcher's perspective: the effects of extrinsic and intrinsic factors on motivation In C. Sansone, and J.M., Harackiewicz (Eds.), Intrinsic and extrinsic motivation: the research for optimal motivation and performance (pp. 309-339). NY: Academic press.

Hidi, S., & Renniger, K. A. (2006). The four-phase model of interest development. *Educational Psychologist*, 41(2), 111-127.

Hoffman, L. (2002). Promoting Girls' Interest and Achievement in Physics Classes for Beginners., 12, 447-465. *Learning and Instruction*, 12, 447-465.

IGDA Report. Game Developer Demographics: An Exploration of Workforce Diversity. October 2005.

Kafai, Y. B. (1994). *Minds in Play: Computer Game Design As a Context for Children's Learning*. Lawrence Erlbaum Associates.

Kekelis, L., Ancheta, R.W., Heber, E., and Countryman, J. 2005. Bridging Differences: How social Relationships and Racial Diversity matter in a Girls' Technology Program. Journal of Women and Minorities in Science and Engineering, 11, 231-246.

Kelleher, C. 2006. Alice: Using 3d gaming technology to draw students into computer science. In *Proceedings of the Fourth Game Design and Technology Workshop and Conference (GDTW 2006)*, 16--20.

Krapp, J.S., Hidi, S., and Renninger, K.A. (1992). Interest, learning, and development. In K.A. Renninger, S. Hidi, and A. Krapp (Eds.), *The role of interest in learning and development* (pp. 3-25). Hillsdale, N.J: Erlbaum.

Lenhardt, A., & Madden, M. (2005). *Teen Content Creators and Consumers.*Washington, DC: Pew Internet & American Life Project, November 2.Available online at http://www.pewInternet.org/PPF/r/166/report_display.asp

Malone, T. (1984). Heuristics for designing enjoyable user interfaces: Lessons from computer games. In J.C. Thomas and M.L. Schneider (Rds.), *Human Factors in Computer Systems*. Norwood, N.J.: Ablex, 1-12.

Maloney, J., L. Burd, Kafai, Y., Rusk, N., Silverman, B., and Resnick, M. (2005). Scratch: A Sneak Preview. International Conference on Creating, Connecting, and Collaborating through Computing., Kyoto, Japan.

Margolis, J., and Fisher, A. (2002). *Unlocking the Clubhouse.* Massachusetts, England: The MIT Press. Cambridge, MA.

Mitchell, M. (1993). Situational interest: Its multifaceted structure in the secondary school mathematics classroom. *Journal of Educational Psychology* 85(3), 424- 436.

Natale, M. (2002). The effect of a male-oriented computer gaming culture on careers in the computer industry. *Computers and society, 24-31.*

National center for women in information technology (NCWIT). (2006). By the Numbers: statistics about women & IT. *NCWIT web site*. Retrieved April 2007, From http://www.ncwit.org/pdf/2006_NCWIT_Statistics.pdf

National report on netday's 2005 speak up event project tomorrow-netday (2006). National report on Our voices, our future. Students and teacher views on science technology & education. NetDay's 2005 Speak Up Event. April 2006, project tomorrow.

Ogazalek, V. (1989). A comparison of male and female computer science students' attitudes toward computers. *SIGCSE Bulletin*, 21(2), 8-14.

Oliver, k (1999). Situated cognition & cognitive apprenticeship. Retrieved July 5 2006 http://www.edtech.vt.edu/edtech/id/models/cog.html

Orr, G., and Nord, M. (2005). A new collaborative teaching model applied to digital music and 3D computer animation. Paper presented at CCSC: Northwestern Conference, 246-253.

Pajares, F., Cheong, Y.F., and Oberman, P. (2004). Psychometric Analysis of Computer Science Help- Seeking scales. Educational and Psychological Measurement, 64 (3), 496-513.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books; 2nd edition.

Pintrich, P. and Schunk D. (1996). *Motivation in education.* Prentice Hall. New Jersey, 301-303.

Renninger, A.K. (2000). Individual interest and its implications for understanding intrinsic motivation. In C. Sansone, and J.M., Harackiewicz (Eds.), *Intrinsic and extrinsic motivation : the research for optimal motivation and performance* (pp. 373-404). NY: Academic press.

Repenning, A., and J. Ambach. (1996). Tactile Programming: A Unified Manipulation Paradigm Supporting Program Comprehension, Composition and Sharing, Proceedings of the 1996 IEEE Symposium of Visual Languages, Boulder, CO, Computer Society, 1996, pp. 102-109.

Repenning, A., and Citrin, W. (1993). Agentsheets: Applying Grid-Based Spatial Reasoning to Human-Computer Interaction, 1993 IEEE Workshop on Visual Languages, Bergen, Norway, IEEE Computer Society Press, 1993, pp. 77-82.

Repenning, A. and Ioannidou, A. (2008). Broadening Participation through Scalable Game Design, ACM Special Interest Group on Computer Science Education Conference, (SIGCSE 2008), (Portland, Oregon USA), ACM Press.

Ryan, R. & Deci, E. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychology*, 55(1), 68-78.
Ryan, R., and Grolnick, W. S. (1986). Origins and pawns in the classroom: Self-report and projective assessments of individual differences in children's perceptions. *Journal of Personality and Social Psychology,* 5(2), 550-558.

Ryan, Stiller, and Lynch (1994). Representations of Relationships to Teachers, Parents, and Friends as Predictors of Academic Motivation and Self-Esteem. The Journal of Early Adolescence, 14(2), 226-249.

Rodger, S.H. (2002). Introducing computer science through animation and virtual worlds Technical Symposium on Computer Science Education. Proceedings of the 33rd SIGCSE technical symposium on Computer science education. Cincinnati, Kentucky. 186 – 190.

Sansone, C., and Morgan, C. (1992). Intrinsic Motivation and Education: Competence in Context. Motivation and Emotion, 16 (3), 249-270.

Schoenfeld, A.H. (1985). *Mathematical Problem Solving.* New York: Academic Press.

Schoenfeld, A.H. (1983). "Problem Solving in the Mathematics Curriculum: A Report, Recommendations and an Annotated Bibliography." The Mathematical Association of America, *MAA Notes,* No. 1.

Schwartz, J., Stagner, J., and Morrison, W. (2006, August). Kid's Programming Language (KPL). *Proceedings of SIGGRAPH 2006*

Seattle Smartgirls workshop (2005). Retrieved June 10, 2005, from http://www.smart-girls.org/seyh/

Smith, F. (2006). Learning from the pros: Animation students link up with professionals for modern-day mentoring. Retrieved October 16, 2006, from http://www.edutopia.org/magazine/ed1article.php?id=Art_1579&issue=jul_06

Squeakland. (2008). Retrieved May 2, 2008, from http://www.squeakland.org/

Steckler, M., & Oleson, K. (2005). SPSS guide. Retrieved May 06, 2007, from http://academic.reed.edu/psychology/RDDAwebsite/spssguide/anova.html

Sung, K., Panitz, M., Wallace, S., Anderson, R., and Nordlinger, J. (2008). Game-Themed Programming Assignments: The Faculty Perspective. In Proceedings of *SIGCSE'08,* Portland, Oregon, 300-304.

Thompson, A. (2008) Computer Science Teacher - Thoughts and Information from Alfred Thompson. Retrieved May 2, 2008, from http://blogs.msdn.com/alfredth/archive/2008/04/09/bootstrapping-xna-into-high-school-computer-science.aspx

Turner, J. C., Meyer, D. K., Cox, K. E., Logan, C., DiCintio, M., & Thomas, C. (1998). Creating Contexts for Involvement in Mathematics. *Journal of Educational Psychology*, 90, 730-745.

Vegso, J. (2007). Continued Drop in CS Bachelor's Degree Production and Enrollments as the Number of New Majors Stabilizes. Computing Research News, 19 (2).

Wenger, E., McDermont, R., & Snyder, W. M. (2002). *Cultivating communities of practice*. Boston, MA: Harvard Business School Press.

Whittington, J. (2003). The process of effective critiques. International Conference on Computer Graphics and Interactive Techniques ACM SIGGRAPH 2003 Educators Program.

Winn, W. (1994). Why I don't want to be an expert sitar player. *Educational Technology*, 34 (8), 11-14.

Appendix A: Student Surveys

Student Demographic Survey

Please complete the following personal information and follow the instructions to create your nickname to answer the surveys. I will use your personal information only to match all your surveys with your art work, programming and my notes during the data collection process.

Personal Information	
First Name	
Last Name	
Email	
Gender	Male __ Female __
Age	
School you attend	
School Grade you will be next fall	9th ____ 10th ____ 11th ____ Other___

To create your own nickname, use the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221

NICKNAME	

Survey 1: Interest in graphic technologies

Thank you for answering this questionnaire. Please complete the following information about your interest in graphic technologies and the way you use it.

NICKNAME ______________

(REMEMBER: your NICKNAME was composed by the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221)

Check the appropriate answer about your experience with digital games.

INTEREST IN COMPUTER GAMES

5. How much do you enjoy playing computer games? (Circle the number).

Not Much A lot

____

1 2 3 4 5

6. How strong is your interest in creating computer games? (Circle the number).

Not too strong Very Strong

____

1 2 3 4 5

7. Have you created any type of game before this camp? If yes, which type? Paper based, computer based, others?

____

8. Do you have a favorite computer game? If yes, can you name it?

____

9. Do you know how computer games are created?

☐ Yes Can you briefly explain what it is needed to create a game? ____________________

☐ No

Check the appropriate answer about your interest in animation

INTEREST IN ANIMATION

1. **How much do you enjoy watching animations? (Circle the number).**

Not Much   A lot

____

1   2   3   4   5

2. **How strong is your interest in creating animations? (Circle the number).**

Not Much   A lot

____

1   2   3   4   5

3. **Have you created an animation before this camp? If yes, which type? Paper based, computer based, others?**

____

____

4. **Do you have a favorite animation? If yes, can you name it?**

____

Check the appropriate answer about your interest in math, art , and computers

INTEREST IN MATH, ART AND COMPUTERS

10. **How good do you think you are at math? Circle the number. (Circle the number).**

Not Good   Very Good

1 2 3 4 5

11. How much do you enjoy doing math? Circle the number. (Circle the number).

Not too much A lot

1 2 3 4 5

12. How strong is your knowledge about using computers? (Circle the number).

Not too strong Very Strong

1 2 3 4 5

13. How much do you enjoy working with computers? (Circle the number).

Not Good Very Good

1 2 3 4 5

14. How good do you think you are at drawing?

Not Good Very Good

1 2 3 4 5

15. How much do you enjoy drawing?

Not too much A lot

1 2 3 4 5

Check the appropriate answer about your interest in the summer camp

INTEREST IN THE SUMMER CAMP

16. What are you most interested to learn in this summer camp?

____

____

17. How did you hear about the summer camp? (mark all that apply)

Friend ____

Family ____

Teacher ____

Advertisement ____

Other ________________

18. What attracted you the most about this summer camp?

____

19. Have you participated in any technology or art related summer camp before?

☐ Yes   Can you name it? ______________________

☐ No

Check the appropriate answer about your experience with computers

ACCESS TO COMPUTERS AND USE OF IT AT HOME

20. Did you have access to a computer at home while you were in: (mark all that apply)

☐ Pre-elementary school

☐ Elementary school

☐ Middle School

☐ High School

21. What do you use the computer for? (mark all that apply)

☐ Do homework

☐ Talk to friends and family (emails, chats, instant messaging)

☐ Play games

☐ Watch DVDs

☐ Surf the Internet

☐ Do programming

☐ Use art software

☐ Other ______________________

22. Have you taking any classes to learn about computers (general computing, programming, art). If yes, please specify the class you have taken

____

____

23. Do you have any close family members, friends or teachers to help you with questions about technology?.

☐ Yes

☐ No

If yes, please mark all that apply: Family___ Friend ____ Teacher _____

24. Do you have access to a computer at home? PC or Mac? Internet?

☐ Yes ☐ PC ☐ MAC ☐ Internet

☐ No

25. Do you have access to a computer outside your home?

☐ Yes ☐ PC ☐ MAC ☐ Internet

☐ No

Survey 2: Programming Self-concept Questionnaire

Thank you for answering this questionnaire. Please complete the following information

NICKNAME _______________

(REMEMBER: your NICKNAME was composed by the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221)

Check the appropriate answer about your thoughts about computer programming

1. How comfortable does programming make you feel? (Circle the number).

Not too comfortable — Very comfortable

1 2 3 4 5

2. How interesting and challenging do you find programming? (Circle the number).

Not too much — A lot

1 2 3 4 5

3. How good are you at programming? (Circle the number).

Not very good — Very good

1 2 3 4 5

4. Are you excited about programming?

Not too excited — Very excited

1 2 3 4 5

5. Do your friends come to you for help with programming?

Not too often — Very often

1 2 3 4 5

6. How well do you do on assignments that require programming?

Not well Very well

1 2 3 4 5

7. Do you have trouble understanding things related to programming?

Not much trouble A lot of trouble

1 2 3 4 5

8. How easy is learning how to program to you?

Not easy Very easy

1 2 3 4 5

Survey 3: Feedback and interest changes

Thank you for answering this questionnaire. Please complete the following information

NICKNAME ________________

(REMEMBER: your NICKNAME was composed by the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221)

Check the appropriate answer about your experience during the summer camp.

SUMMER CAMP EXPERIENCE

1. How much did you enjoy the summer camp? (Circle the number).

Not Much ... A lot

1 2 3 4 5

2. What did you enjoy the most about the camp?

____

____

3. What was your favorite activity in the camp? Why?

____

____

4. Would you recommend the summer camp to your friends who are interested in games and animation?

☐ Yes

☐ No

☐ Don't know

5. How the things you did in the camp fit with what you want to do in the future as a career or hobby?

____

6. How much did you like your summer camp classmates?

Not too much ... A lot

1 2 3 4 5

7. Will you keep in touch with people you met in the camp?

☐ Yes

☐ No

☐ Maybe

8. How much did you like your summer camp instructor?

Not too much A lot

1 2 3 4 5

10. How much the things you did in the camp fit with what you want to do in the future as a career or hobby?

Not too much A lot

1 2 3 4 5

INTEREST IN GAMES AND ANIMATION

11. Now that you learned how computer games are created, can you briefly explain what it is needed to create a game?

____

____

12. How strong is your current interest in creating computer games? (Circle the number).

Not too strong Very Strong

1 2 3 4 5

13. How strong is your current interest in creating animations? (Circle the number).

Not too strong Very Strong

1 2 3 4 5

14. Which way of learning about games and animations did you like the most? (mark all that apply)

☐ Working in groups ☐ Step by Step tutorials

☐ Working alone ☐ Working at your own pace

☐ Lectures ☐ Interacting with your classmates in the classroom

☐ Guest speaker presentations ☐ Choosing the activity you want to work on (art or programming)

15. Do you want to learn more about a particular topic related to games and animations?

____

____

THINGS YOU DID OUTSIDE THE SUMMER CAMP

16. Did you have the opportunity to talk about what you did in the camp with friends and family?

☐ Yes

☐ No

If yes, please mark all that apply: Family____ Friend ____ Teacher _____

17. How often did you work on your final project at home?

☐ Frequently

☐ Occasionally

☐ Rarely OR

☐ Never

18. Did you use additional resources (web sites, books, help from friends and family, etc) to complete your project beside what it was provided to you in the summer camp?

☐ Yes What kind of resoources?:____

☐ No

19. Are you planning on continue working on your project at home or school?

☐ Yes

☐ No

☐ Maybe

20. Did you do the extra assignment proposed by the instructor?

☐ Yes

☐ No

If yes

a. How long did it take you to complete the extra assignment? ____________

b. What kind of help did you use to complete the assignment? ____________

☐ Online video class tutorials

☐ Chat with teaching assistant

☐ External online tutorials

☐ Books

☐ Others: ____________________

ABOUT THE CLASS WEBSITE

21. What was helpful about the class website?

____

____

22. How often did you use the class website?

☐ Frequently

☐ Occasionally

☐ Rarely OR

☐ Never

23. What did you use the class website for?

____

____

24. Will you be using the class website in the future?

☐ Yes

☐ No

☐ Maybe

Survey 4: Post summer camp survey

Thank you for answering this questionnaire. Please complete the following information

NICKNAME _______________

(REMEMBER: your NICKNAME was composed by the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221)

Check the appropriate answer about your experiences with games and animation technology after the camp

1. Did you continue working in the final project you did in the summer camp after the camp was over?

☐ Yes

☐ No

2. Have you created any new game or animation beside the one you created in the summer camp?

☐ Yes

☐ No

3. How often have you used the class web site in the past two months?

☐ Frequently

☐ Occasionally

☐ Rarely OR

☐ Never

4. Do you keep in touch with some of the people you met at the summer camp?

☐ Yes

☐ No

5. Have you voluntarily taken any classes related to computing, digital art or math?

☐ Yes, What classes?: ________________________________

☐ No

6. How much access to computers and internet you have in the past two months?
Computers:

Not Much A lot

____

1 2 3 4 5

Internet:

Not Much A lot

1 2 3 4 5

7. Have you read any books or magazines related to games, animation, or art, in the past two months?

☐ Yes Can you name it: ______________________________

☐ No

8. How often do you visit web sites devoted to games or animation?

☐ Frequently

☐ Occasionally

☐ Rarely OR

☐ Never

9. How much encouragement have you received from your teachers in school to continue learning about technology?

Not Much A lot

1 2 3 4 5

10. How strong is your current interest in creating computer games? (Circle the number).

Not too strong Very Strong

1 2 3 4 5

11. How strong is your current interest in creating animations? (Circle the number).

Not too strong Very Strong

1 2 3 4 5

Appendix B: Teacher Surveys

Teacher Demographic Survey

Please complete the following personal information and follow the instructions to create your nickname to answer the surveys. I will use your personal information only to match all your surveys with your art work, programming code and my notes during the data collection process.

Personal Information	
First Name	
Last Name	
Email	
Gender	Male __ Female __
School where you teach	
Grade you teach	9th ____ 10th ____ 11th ____ 12th ____
Subject(s) you teach	

To create your own nickname, use the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221

NICKNAME	

Survey 1: Interest in graphic technologies

Thank you for answering this questionnaire. Please complete the following information about your interest in graphic technologies and the way you use it.

NICKNAME ______________

(REMEMBER: *your NICKNAME was composed by the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221)*

Check the appropriate answer about your experience with digital games.

INTEREST IN COMPUTER GAMES

1. How much do you enjoy playing computer games? (Circle the number).

Not Much   A lot

1 2 3 4 5

2. How strong is your interest in creating computer games to teach math and science? (Circle the number).

Not too strong   Very Strong

1 2 3 4 5

3. Have you created any type of game before this summer institute? If yes, which type? Paper based, computer based, others?

____

____

4. Do you have a favorite computer game? If yes, can you name it?

____

5. Do you know how computer games are created?

□ Yes Can you briefly explain what it is needed to create a game? ______________________

□ No

Check the appropriate answer about your interest in animation

INTEREST IN ANIMATION

6. How much do you enjoy watching animations? (Circle the number).

Not Much A lot

1 2 3 4 5

7. How strong is your interest in creating animations to teach math and science? (Circle the number).

Not Much A lot

1 2 3 4 5

8. Have you created an animation before this summer institute? If yes, which type? Paper based, computer based, others?

____

____

____

9. Do you have a favorite animation? If yes, can you name it?

____

Check the appropriate answer about your interest in math, art , and computers

INTEREST IN MATH, ART AND COMPUTERS

10. How good do you think you are at teaching math, science and/or technology?. (Circle the number).

Not Good Very Good

1 2 3 4 5

11. How much do you enjoy teaching math, science and/or technology?. (Circle the number).

Not too much A lot

1 2 3 4 5

12. How strong is your knowledge about using computers? (Circle the number).

Not Good Very Good

1 2 3 4 5

13. How much do you enjoy working with computers? (Circle the number).

Not Good Very Good

1 2 3 4 5

14. How good do you think you are at drawing?

Not Good Very Good

1 2 3 4 5

15. How much do you enjoy drawing?

Not much A lot

1 2 3 4 5

Check the appropriate answer about your interest in the summer institute

INTEREST IN THE SUMMER INSTITUTE

16. What are you most interested to learn in this summer institute?

____

____

____

17. How did you hear about the summer institute? (mark all that apply)

Friend ____

Colleague ____

Family ____

Advertisement ____

Other ____________

18. What attracted you the most about this summer institute?

____

19. Have you participated in any technology or art related institute/workshop before?

☐ Yes Can you name it? ______________________

☐ No

Check the appropriate answer about your experience with computers

ACCESS TO COMPUTERS AND USE OF IT AT HOME

20. Did you have access to computers while you were in: (mark all that apply)

☐ Primary school

☐ Secondary School

☐ College

21. What do you use the computer for? (mark all that apply)

☐ Prepare classes for my students

☐ Write reports

☐ Use productivity tool (word processors, spread sheets, PowerPoint, etc)

☐ Talk to friends and family (emails, chats, instant messaging)

☐ Share pictures with family and friends

☐ Play games

☐ Watch DVDs

☐ Surf the Internet

☐ Do programming

☐ Use art software

☐ Other ________________________

22. Have you taking any classes to learn about computers (general computing, programming, art). If yes, please specify the class you have taken

____

____

23. Do you have any close family members, friend or coworker to help you with questions about technology?.

☐ Yes

☐ No

If yes, please mark all that apply: Family___ Friend ____ coworker _____

24. Do you have access to a computer at home? PC or Mac? Internet?

☐ Yes ☐ PC ☐ MAC ☐ Internet

☐ No

25. Do you have access to a computer outside your home?

☐ Yes ☐ PC ☐ MAC ☐ Internet

☐ No

Survey 2: Programming Self-concept Questionnaire

Thank you for answering this questionnaire. Please complete the following information

NICKNAME _______________

(REMEMBER: *your NICKNAME was composed by the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221)*

Check the appropriate answer about your thoughts about computer programming

1. How comfortable does programming make you feel? (Circle the number).

Not too comfortable — Very comfortable

1 2 3 4 5

2. How interesting and challenging do you find programming? (Circle the number).

Not too much — A lot

1 2 3 4 5

3. How good are you at programming? (Circle the number).

Not very good — Very good

1 2 3 4 5

4. Are you excited about programming?

Not too excited — Very excited

1 2 3 4 5

5. Do your colleagues come to you for help with programming?

Not too often — Very often

1 2 3 4 5

6. How well do you do on tasks that require programming?

Not well				Very well
1	2	3	4	5

7. Do you have trouble understanding things related to programming?

Not much trouble				A lot of trouble
1	2	3	4	5

8. How easy is learning how to program for you?

Not easy				Very easy
1	2	3	4	5

Survey 3: Feedback and interest changes

Thank you for answering this questionnaire. Please complete the following information

NICKNAME ______________

(REMEMBER*: your NICKNAME was composed by the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221)*

Check the appropriate answer about your experience during the summer institute.

SUMMER INSTITUTE EXPERIENCE

1. How much did you enjoy the summer institute? (Circle the number).

Not Much A lot

____

1 2 3 4 5

2. What did you enjoy the most about the institute?

____

____

3. What was your favorite activity in the summer institute? Why?

____

____

4. Would you recommend the summer institute to your coworkers?

☐ Yes

☐ No

☐ Maybe

5. How the things you did in the camp fit with what you want to teach in your school?

____

____

6. How much did you like your summer institute colleagues?

Not too much A lot

1 2 3 4 5

7. Will you keep in touch with people you meet in the summer institute?

☐ Yes

☐ No

☐ Maybe

8. How confident are you that you will be able to teach to your students what you learned in the teacher institute? (Select one circle)

Not too much A lot

1 2 3 4 5

9. How appropriate was the content/skills you learned in the teacher institute to teach math, science or computing to your students? (Select one circle)

Not too appropriate Very appropriate

1 2 3 4 5

INTEREST IN GAMES AND ANIMATION

11. Now that you learned how computer games are created, can you briefly explain what it is needed to create a game?

12. How strong is your current interest in creating computer games to teach math and science? (Circle the number).

Not too strong   Very Strong

---

1   2   3   4   5

13. How strong is your current interest in creating animations to teach math and science? (Circle the number).

Not too strong   Very Strong

---

1   2   3   4   5

14. Which way of learning about games and animations did you like the most? (mark all that apply)

- ☐ Working in groups
- ☐ Step by Step tutorials
- ☐ Working alone
- ☐ Working at your own pace
- ☐ Lectures
- ☐ Interacting with your colleagues in the classroom
- ☐ Guest speaker presentations
- ☐ Choosing the activity you want to work on (art or programming)
- ☐ Getting tips about how to teach the content I am learning
- ☐ Hearing from colleagues about what they are doing in their schools to teach math/science/computing

15. Do you want to learn more about a particular topic related to games and animations?

____

____

THINGS YOU DID OUTSIDE THE SUMMER INSTITUTE

16. Did you have the opportunity to talk about what you did in the institute with friends, family, or coworkers?.

☐ Yes

☐ No

If yes, please mark all that apply: Family___ Friend ____ coworker _____

17. How often did you work on your final project at home?

☐ Frequently

☐ Occasionally

☐ Rarely OR

☐ Never

18. Did you use additional resources (web sites, books, help from friends and family, etc) to complete your project beside what it was provided to you in the summer institute?

☐ Yes

What kind of resources?:____

☐ No

19. Are you planning on continue working on your project at home or school?

☐ Yes

☐ No

☐ Maybe

20. Did you do the extra activity proposed by the instructor?

☐ Yes

☐ No

If yes

a. How long did it take you to complete the extra activity? ____________

b. What kind of help did you use to complete the activity?

☐ Online video class tutorials

☐ Chat with teaching assistant

☐ External online tutorials

☐ Books

☐ Others: ___________________

ABOUT THE CLASS WEBSITE

22. What was helpful about the class website?

____

____

23. How often did you use the class website?

☐ Frequently

☐ Occasionally

☐ Rarely OR

☐ Never

24. What did you use the class website for?

____

____

25. Will you be using the class website in the future?

☐ Yes

☐ No

☐ Maybe

Survey 4: Post summer institute survey

Thank you for answering this questionnaire. Please complete the following information

NICKNAME ______________

(REMEMBER: *your NICKNAME was composed by the first 3 letters of your favorite animal and the month and day of your birthday. Example: If my favorite animal is "bear" and my birthday is December 21, the nickname would be BEA1221)*

Check the appropriate answer about your experiences with games and animation technology after the institute

1. Did you continue working in the final project you did in the summer institute after it was over?

☐ Yes

☐ No

2. Have you created any new game or animation beside the one you created in the summer institute?

☐ Yes

☐ No

3. How often have you used the class web site in the past two months?

☐ Frequently

☐ Occasionally

☐ Rarely OR

☐ Never

4. Do you keep in touch with some of the people you met at the summer institute?

☐ Yes

☐ No

5. Have you used what you learned in the teacher institute to teach math/science or technology?

☐ Yes, What classes?: ______________________________

☐ No

6. How much access to computers and internet you have in the past two months?

Computers:

Not Much A lot

1 2 3 4 5

Internet:

Not Much A lot

1 2 3 4 5

7. Have you read any books or magazines related to games, animation, or art, in the past two months?

☐ Yes Can you name it: ______________________________

☐ No

8. How often do you visit web sites devoted to games or animation?

☐ Frequently

☐ Occasionally

☐ Rarely OR

☐ Never

9. How much positive feedback and outcomes have you received from your students about using games and animation to teach?

Not Much A lot

1 2 3 4 5

10. How strong is your current interest in creating computer games to teach math and science? (Circle the number).

Not too strong Very Strong

1 2 3 4 5

11. How strong is your current interest in creating animations to teach math and science? (Circle the number).

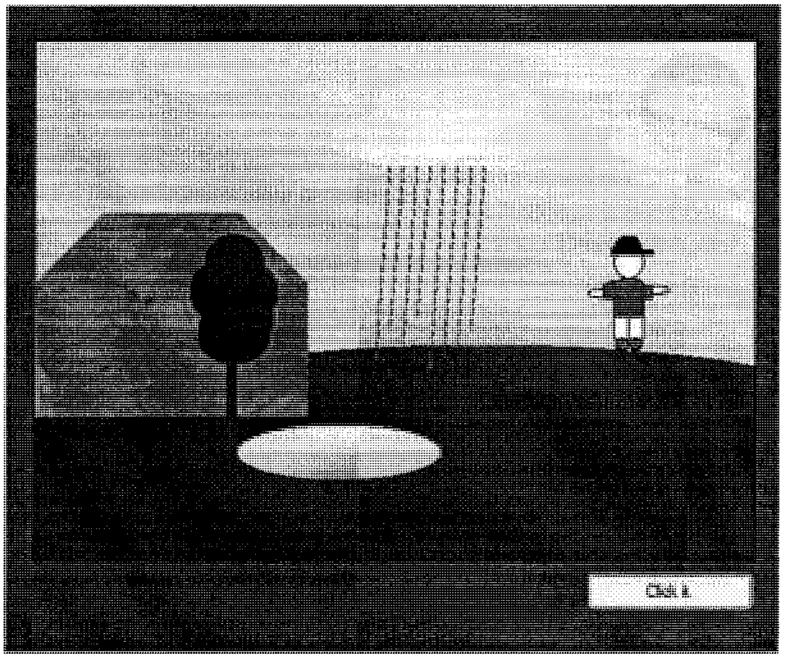
Not too strong Very Strong

1	2	3	4	5

Appendix C: Sample Programming Code

(Students)

Visual Basic Programming Task 1: Create a scene that contains a house, lake, skyline, sun, clouds, a character and rain. Use different texture brushes and geometric shapes.

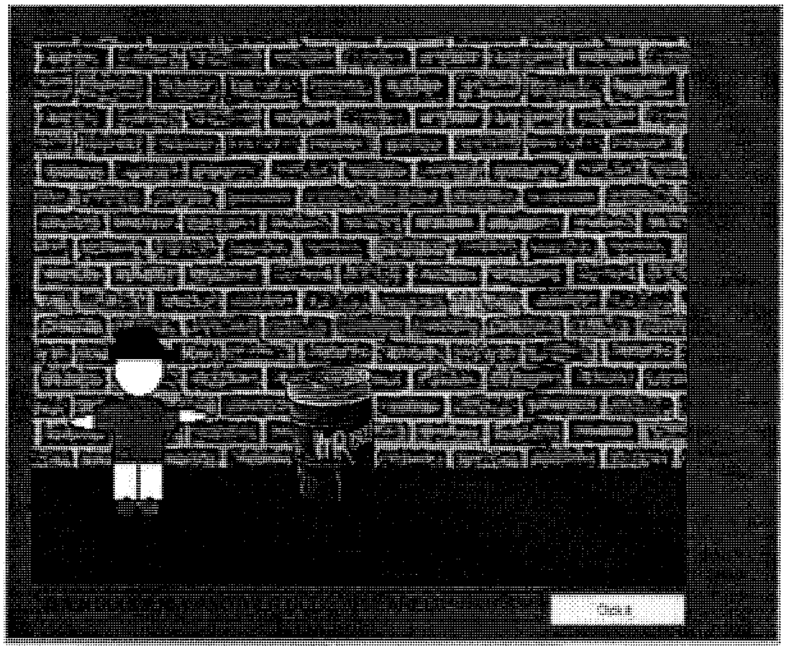


```
Imports System.Drawing.Drawing2D
Public Class Form1
   Private canvas As Graphics
   'Private pointList(2) As Point
   Private customBrush As Brush
   Private cloudBrush As Brush
   Private patternBrush As Brush
   Private sunriseBrush As Brush
   Private waterBrush As Brush
   Private sprite As Bitmap
   Private cloudBrush1 As Brush
   Private rainPen As Pen
   Private pointList(5) As Point
   Private brickTexture As Bitmap
   Private brickBrush As Brush
   Private pointList1(3) As Point
   Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click
```

```
    canvas = picAnimation.CreateGraphics
    canvas.FillRectangle(Brushes.LightBlue, 0, 0, 650, 275)
    sunriseBrush = New LinearGradientBrush(New Rectangle(0, 0, 650, 300),
Color.White, Color.LightSkyBlue, LinearGradientMode.BackwardDiagonal)
    canvas.FillRectangle(sunriseBrush, 0, 0, 650, 300)
    'sun
    cloudBrush = New SolidBrush(Color.FromArgb(150, 255, 255, 255))
    canvas.FillEllipse(Brushes.Yellow, 444, 8, 80, 80)
    canvas.FillEllipse(cloudBrush, 420, 60, 90, 30)
    canvas.FillEllipse(cloudBrush, 450, 30, 85, 27)

    customBrush = New SolidBrush(Color.FromArgb(89, 75, 20))

    'hill
    canvas.FillEllipse(Brushes.Green, 0, 220, 655, 150)

    'ground
    patternBrush = New HatchBrush(HatchStyle.DarkUpwardDiagonal,
Color.DarkGreen, Color.Green)
    canvas.FillRectangle(patternBrush, 0, 275, 650, 400)
    'house
    'pointList(0) = New Point(65, 150)
    'pointList(1) = New Point(15, 200)
    'pointList(2) = New Point(115, 200)
    'house structure
    pointList(0) = New Point(50, 125)
    pointList(1) = New Point(150, 125)
    pointList(2) = New Point(200, 175)
    pointList(3) = New Point(200, 275)
    pointList(4) = New Point(0, 275)
    pointList(5) = New Point(0, 175)

    brickTexture = New Bitmap("slates.jpg")
    brickBrush = New TextureBrush(brickTexture)
    canvas.FillPolygon(brickBrush, pointList)

    'house roofing

    'canvas.FillRectangle(Brushes.AntiqueWhite, 15, 200, 100, 100)
    'canvas.FillRectangle(Brushes.DarkSlateGray, 48, 260, 30, 40)
```

```
    'tree
    canvas.FillRectangle(Brushes.Brown, 140, 200, 8, 100)
    'canvas.FillEllipse(
    cloudBrush = New SolidBrush(Color.FromArgb(150, 10, 255, 10))
    canvas.FillEllipse(Brushes.DarkGreen, 128, 190, 45, 45)
    canvas.FillEllipse(Brushes.DarkGreen, 130, 180, 45, 45)
    canvas.FillEllipse(Brushes.DarkGreen, 113, 160, 45, 45)
    canvas.FillEllipse(Brushes.DarkGreen, 133, 160, 45, 45)
    canvas.FillEllipse(Brushes.DarkGreen, 123, 140, 45, 45)
    canvas.FillEllipse(Brushes.DarkGreen, 118, 190, 45, 45)
    canvas.FillEllipse(Brushes.DarkGreen, 112, 160, 45, 45)

    waterBrush = New LinearGradientBrush(New Rectangle(148, 280, 150, 40),
Color.White, Color.LightSkyBlue, LinearGradientMode.Vertical)
    canvas.FillEllipse(waterBrush, 148, 280, 150, 40)

    sprite = New Bitmap("animation1.bmp")
    sprite.MakeTransparent(Color.Blue)
    canvas.DrawImage(sprite, 390, 138)

    'DrawCloud
    cloudBrush1 = New SolidBrush(Color.FromArgb(100, 255, 255, 255))
    canvas.FillEllipse(cloudBrush1, 215, 50, 120, 40)
    canvas.FillEllipse(cloudBrush1, 270, 50, 120, 40)
    canvas.FillEllipse(cloudBrush1, 250, 25, 120, 40)

    'rain
    rainPen = New Pen(Color.SlateBlue, 2)
    rainPen.DashStyle = (DashStyle.Dash)
    canvas.DrawLine(rainPen, 260, 90, 250, 230)
    canvas.DrawLine(rainPen, 270, 90, 260, 210)
    canvas.DrawLine(rainPen, 280, 90, 270, 220)
    canvas.DrawLine(rainPen, 290, 90, 280, 200)
    canvas.DrawLine(rainPen, 300, 90, 290, 240)
    canvas.DrawLine(rainPen, 310, 90, 300, 230)
    canvas.DrawLine(rainPen, 320, 90, 310, 225)
    canvas.DrawLine(rainPen, 330, 90, 320, 210)
  End Sub
End Class
```

Visual Basic Programming Task 2: Implement a background scrolling effect where the background image move to the left continuously, add a character to a background image previously created in the Photoshop class, add written storyline or instructions to play the game, add sound, and implement user's keyboard actions.

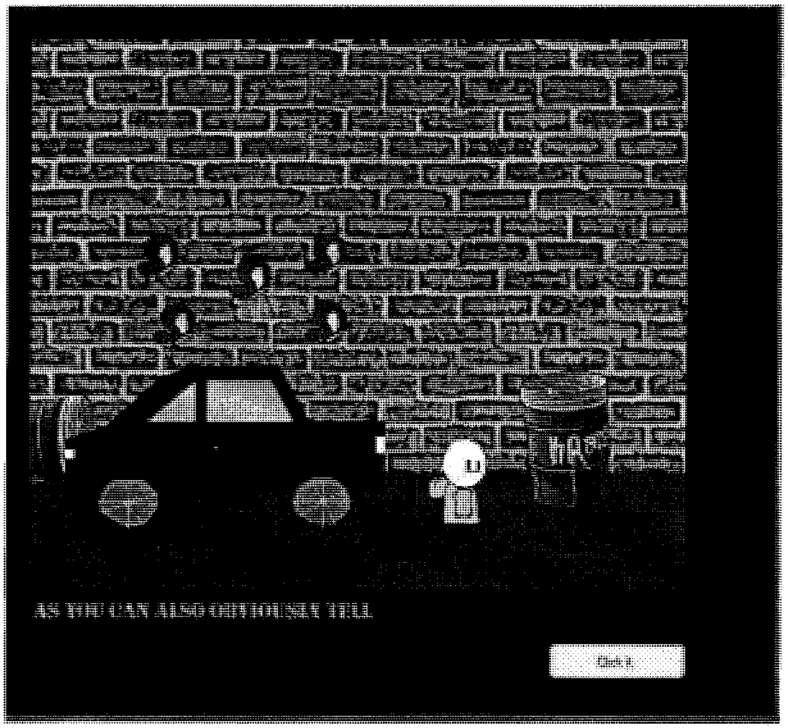


```
Imports System.Drawing.Drawing2D
Imports System.Threading
Public Class Form1
    Private bookMark As Decimal
    Private canvas As Graphics
    Private bgHiddenCanvas As Graphics
    Private spriteHiddenCanvas As Graphics

    Private background As Bitmap
    Private sprite As Bitmap
    Private guy As Bitmap
    Private hiddenImage As Bitmap

    Private x As Integer
    Private xSprite As Integer
    Private ySprite As Integer
```

```
    Private bgscrollMatrix As Matrix
    Private scrollLength As Integer
    Private userQuits As Boolean
    Private isAntag As Boolean

Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click
        canvas = picAnimation.CreateGraphics

        background = New Bitmap("thebackground.bmp")
        sprite = New Bitmap("animation1.bmp")
        sprite.MakeTransparent(Color.Blue)

        guy = New Bitmap("animation1.bmp")
        guy.MakeTransparent(Color.Blue)


        hiddenImage = New Bitmap(picAnimation.Width, picAnimation.Height)
        bgHiddenCanvas = Graphics.FromImage(hiddenImage)
        spriteHiddenCanvas = Graphics.FromImage(hiddenImage)

        bgscrollMatrix = New Matrix(1, 0, 0, 1, -2, 0)
        scrollLength = background.Width - picAnimation.Width

        btnStart.Enabled = False
        mediaPlayer1.URL = "badboys.wav"
        mediaPlayer1.Ctlcontrols.play()

        lblStory.Text = "ONCE UPON A TIME THERE WAS A DUDE
WALKING...OBVIOUSLY"
        timStartMusic.Enabled = True
        isAntag = False
        Do
            bgHiddenCanvas.DrawImage(background, 0, 0, 2100, 500)

            spriteHiddenCanvas.DrawImage(sprite, xSprite, ySprite, 210, 210)

            canvas.DrawImage(hiddenImage, 0, 0)

            Thread.Sleep(15)
```

```
            bgHiddenCanvas.MultiplyTransform(bgscrollMatrix)
            x = x + 2

            If x >= scrollLength Then
                x = 0
                bgHiddenCanvas.ResetTransform()
            End If

            Application.DoEvents()
        Loop Until userQuits
        End Sub

    Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
    System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
        userQuits = True
    End Sub

    Private Sub Form1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Me.KeyDown
        If e.KeyCode = Keys.Up Then
            ySprite = ySprite - 15
        End If
        If e.KeyCode = Keys.Down Then
            ySprite = ySprite + 15
        End If
        If e.KeyCode = Keys.Left Then
            xSprite = xSprite - 10
        End If
        If e.KeyCode = Keys.Right Then
            xSprite = xSprite + 10
        End If
    End Sub
    Private Sub timStartMusic_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles timStartMusic.Tick
        lblStory.Text = "THEN SUDDENLY ANOTHER DUDE APPEARED... "
        timStartMusic.Enabled = False
        timAntagonist.Enabled = True
        mediaPlayer1.URL = "glassbrk.wav"
        isAntag = True
    End Sub
End Class
```

Visual Basic Programming Task 3 (Final Project): Create an animation or game, preferably that was a more complete version of what they did for their programming task 2




```
Imports System.Drawing.Drawing2D
Imports System.Threading
'Butterflies follow a guy who eventually gets runned over by car
'Theme: Bad luck
Public Class Form1
    Private bookMark As Decimal
    Private canvas As Graphics
    Private bgHiddenCanvas As Graphics
    Private spriteHiddenCanvas As Graphics

    Private background As Bitmap
    Private sprite As Bitmap
    Private guy As Bitmap
    Private hiddenImage As Bitmap
```

```
'Sprite animation variables
Private frameWidth As Integer = 225
Private frameHeight As Integer = 135
Private frameX As Integer
Private frameRectangle As Rectangle
Private destinationRectangle As Rectangle
Private loopCount As Integer

'gray guy
Private frameWidth1 As Integer = 217
Private frameHeight1 As Integer = 100
Private frameX1 As Integer
Private frameRectangle1 As Rectangle
Private destinationRectangle1 As Rectangle

'gray2
Private gray2 As Bitmap
Private frameWidth2 As Integer = 94
Private frameHeight2 As Integer = 100
Private frameX2 As Integer
Private frameRectangle2 As Rectangle
Private destinationRectangle2 As Rectangle

'Car
Private car As Bitmap
Private seeCar As Boolean = False
Private xCarmove As Integer = -200
Private dies As Boolean = False
Private appears As Boolean = True
Private dead As Bitmap
Private x As Integer
Private xSprite As Decimal
Private ySprite As Decimal
Private xSpriteSpeed As Decimal
Private ySpriteSpeed As Decimal
Private bgscrollMatrix As Matrix
Private scrollLength As Integer
Private scrollSpeed As Integer
Private userQuits As Boolean
Private showDesert As Boolean
Private noCycles As Integer
```

```
   Private gray As Bitmap
   Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click
      canvas = picAnimation.CreateGraphics
      lblStory.Text = "ONCE UPON A TIME THERE WAS A DUDE WALKING..."
      mediaPlayer1.URL = "panther1.wav"
      background = New Bitmap("thebackground.bmp")
      gray = New Bitmap("gray.bmp")
      gray.MakeTransparent(Color.Blue)
      gray2 = New Bitmap("graytwo.bmp")
      gray2.MakeTransparent(Color.Blue)
      car = New Bitmap("car.bmp")
      car.MakeTransparent(Color.Blue)
      dead = New Bitmap("graydead.bmp")
      dead.MakeTransparent(Color.Blue)
      sprite = New Bitmap("butterfly.bmp")
      sprite.MakeTransparent(Color.Blue)
      hiddenImage = New Bitmap(picAnimation.Width, picAnimation.Height)
      bgHiddenCanvas = Graphics.FromImage(hiddenImage)
      spriteHiddenCanvas = Graphics.FromImage(hiddenImage)
      scrollSpeed = 2
      bgscrollMatrix = New Matrix(1, 0, 0, 1, -scrollSpeed, 0)
      scrollLength = background.Width - picAnimation.Width
      btnStart.Enabled = False
      'mediaPlayer1.URL = "birdsynth.mp3"
      'mediaPlayer1.Ctlcontrols.play()
      timCarPass.Enabled = True
      TimLookBack.Enabled = True
      timStartMusic.Enabled = True
      TimCrash.Enabled = True
      TimChangelbl.Enabled = True
      TimlblChange.Enabled = True
      TimlblChange2.Enabled = True
      TimlblChange5.Enabled = True
      Do
         mediaPlayer1.Ctlcontrols.play()
         xSprite = xSprite + xSpriteSpeed
         ySprite = ySprite + ySpriteSpeed

         'first draw bg
         bgHiddenCanvas.DrawImage(background, 0, 0, 2100, 500)
```

```
        'next:select the current frame
        frameRectangle = New Rectangle(frameX, 0, frameWidth, frameHeight)
        destinationRectangle = New Rectangle(xSprite, ySprite, frameWidth,
frameHeight)
        spriteHiddenCanvas.DrawImage(sprite, destinationRectangle, frameRectangle,
GraphicsUnit.Pixel)
        'gray guy
        If appears = True Then
          frameRectangle1 = New Rectangle(frameX1, 0, frameWidth1, frameHeight1)
          destinationRectangle1 = New Rectangle(300, 350, frameWidth1,
frameHeight1)
          spriteHiddenCanvas.DrawImage(gray, destinationRectangle1,
frameRectangle1, GraphicsUnit.Pixel)
        End If

        If seeCar = True Then
          spriteHiddenCanvas.DrawImage(car, xCarmove, 240)
          xCarmove = xCarmove + 9
        End If
        'If x >= 600 Then
        '   gray = New Bitmap("graytwo.bmp")
        '   gray.MakeTransparent(Color.Blue)
        'End If

        'If x >= 620 Then
        '   gray = New Bitmap("gray.bmp")
        '   gray.MakeTransparent(Color.Blue)
        'End If

        If dies = True Then
          gray = New Bitmap("grayranover.bmp")
          gray.MakeTransparent(Color.Blue)

        End If
        If xCarmove >= 230 Then
          dies = True
        End If

        If xCarmove >= 250 Then
          dies = False
          appears = False
```

```
                spriteHiddenCanvas.DrawImage(dead, 300, 350, 208, 100)
                scrollSpeed = 0
                bgscrollMatrix = New Matrix(1, 0, 0, 1, -scrollSpeed, 0)
            End If

            'advance to next frame
            If loopCount Mod 8 = 0 Then
                frameX = frameX + frameWidth
                loopCount = 0
            End If

            If loopCount Mod 8 = 0 Then
                frameX1 = frameX1 + frameWidth1
                loopCount = 0
            End If

            If frameX1 >= gray.Width Then
                frameX1 = 0
            End If

            loopCount = loopCount + 1
            If frameX >= sprite.Width Then
                frameX = 0
            End If
            canvas.DrawImage(hiddenImage, 0, 0)
            Thread.Sleep(10)
            bgHiddenCanvas.MultiplyTransform(bgscrollMatrix)
            x = x + scrollSpeed
            If x >= scrollLength Then
                x = 0
                bgHiddenCanvas.ResetTransform()
            End If
            Application.DoEvents()
        Loop Until userQuits
    End Sub
    Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
        userQuits = True
    End Sub
    Private Sub Form1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Me.KeyDown
```

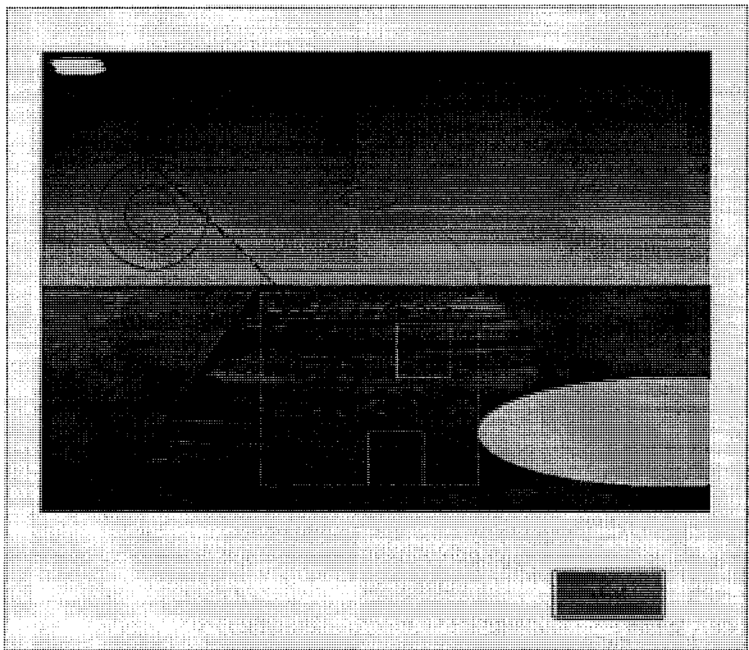
```
        'step mode
        If e.KeyCode = Keys.Up Then
            ySprite = ySprite - 20
        End If
        If e.KeyCode = Keys.Down Then
            ySprite = ySprite + 20
        End If
        If e.KeyCode = Keys.Left Then
            xSprite = xSprite - 20
        End If
        If e.KeyCode = Keys.Right Then
            xSprite = xSprite + 20
        End If
    End Sub
    Private Sub timStartMusic_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles timStartMusic.Tick
        'timStartMusic.Enabled = False
    End Sub
    Private Sub TimLookBack_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimLookBack.Tick
        TimLookBack.Enabled = False
        timCarPass.Enabled = True
    End Sub
    Private Sub timCarPass_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles timCarPass.Tick
        seeCar = True
        timCarPass.Enabled = False
    End Sub
    Private Sub TimCrash_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        mediaPlayer1.URL = "car_crash.wav"
        mediaPlayer1.Ctlcontrols.play()
        TimCrash.Enabled = False
    End Sub
    Private Sub TimCrash_Tick_2(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimCrash.Tick
        mediaPlayer1.URL = "car_crash.wav"
        mediaPlayer1.Ctlcontrols.play()
        TimCrash.Enabled = False
        TimChangeMusicBack.Enabled = True
    End Sub
```

```
    Private Sub TimChangeMusicBack_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TimChangeMusicBack.Tick
        mediaPlayer1.URL = "godfat~1.wav"
        mediaPlayer1.Ctlcontrols.play()
    End Sub
    Private Sub TimChangelbl_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimChangelbl.Tick
        lblStory.Text = "...OBVIOUSLY"
        TimChangelbl.Enabled = False
    End Sub
    Private Sub TimlblChange_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimlblChange.Tick
        lblStory.Text = "AND FIVE BUTTERFLIES WERE FOLLOWING HIM"
        TimlblChange.Enabled = False
    End Sub
    Private Sub TimlblChange2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimlblChange2.Tick
        lblStory.Text = "AS YOU CAN ALSO OBVIOUSLY TELL"
        TimlblChange2.Enabled = False
    End Sub
    Private Sub TimlblChange5_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimlblChange5.Tick
        lblStory.Text = "OH........AND HE DIES"
        TimlblChange5.Enabled = False
    End Sub
End Class
```

Appendix D: Sample Programming Code

(Teachers)

Visual Basic Programming Task 1: Create a scene that contains a house, lake, skyline, sun, clouds, a character and rain. Use different texture brushes and geometric shapes.

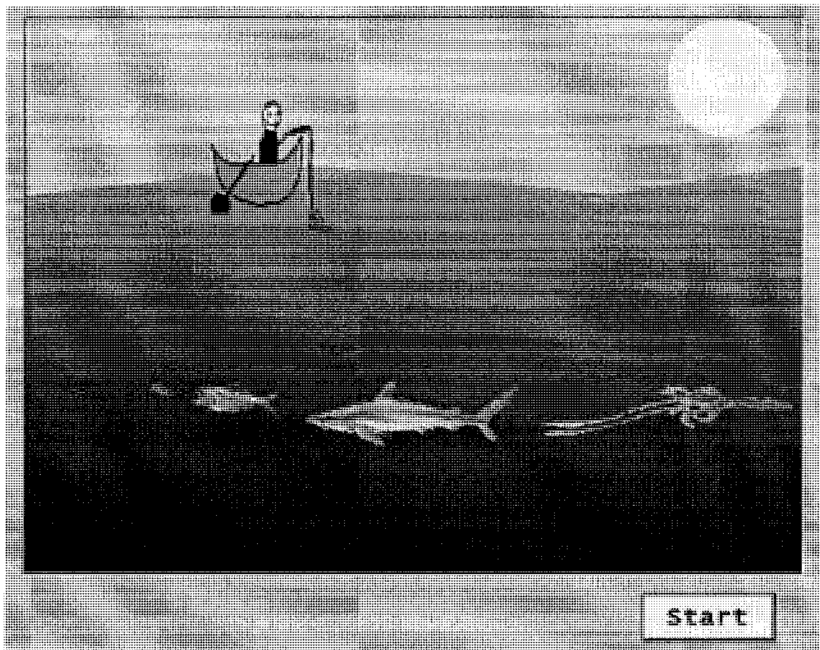


```
Imports System.Drawing.Drawing2D
Public Class Form1
    ' Declaration Section
    Private canvas As Graphics
    Private pointlist(4) As Point
    Private pointlist2(6) As Point
    Private mycolorbrush As Brush
    Private sunsetbrush As Brush
    Private lakebrush As Brush
    Private background As Bitmap
    Private grassbrush As Brush
    Private grasstexture As Bitmap
    Private rainpen As Pen
    Private Sub btnstart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnstart.Click
        Me.BackColor = Color.MistyRose

        canvas = picAnimation.CreateGraphics
        background = New Bitmap("Patrick's dock in Turkey.jpg")
```

```
grasstexture = New Bitmap("GRASS texture.jpg")

'add image
canvas.DrawImage(background, 0, 0, picAnimation.Width, picAnimation.Height)

'draw ground
'create a grass texture brush
'grassbrush = New TextureBrush(grasstexture)

'canvas.FillRectangle(grassbrush, 0, 214, picAnimation.Width,
picAnimation.Height)

'draw sky
sunsetbrush = New LinearGradientBrush(New Rectangle(0, 0,
picAnimation.Width, 250), Color.DarkBlue, Color.SkyBlue,
LinearGradientMode.Vertical)

canvas.FillRectangle(sunsetbrush, 0, 0, picAnimation.Width, picAnimation.Height
\ 2)

'draw mountains
pointlist2(0) = New Point(0, 214)
pointlist2(1) = New Point(100, 400)
pointlist2(2) = New Point(150, 300)
pointlist2(3) = New Point(200, 420)
pointlist2(4) = New Point(250, 250)
pointlist2(5) = New Point(300, 380)
pointlist2(6) = New Point(400, 214)

'canvas.FillPolygon(Brushes.AntiqueWhite, pointlist2)

' house body
canvas.DrawLine(Pens.ForestGreen, 200, 200, 400, 200)
canvas.DrawLine(Pens.ForestGreen, 200, 200, 300, 100)
canvas.DrawLine(Pens.ForestGreen, 300, 100, 400, 200)

' roof triange
canvas.DrawLine(Pens.ForestGreen, 200, 200, 200, 400)
canvas.DrawLine(Pens.ForestGreen, 400, 200, 400, 400)
canvas.DrawLine(Pens.ForestGreen, 200, 400, 400, 400)
```

```
        'door
        canvas.DrawLine(Pens.Fuchsia, 300, 400, 300, 350)
        canvas.DrawLine(Pens.Fuchsia, 300, 350, 350, 350)
        canvas.DrawLine(Pens.Fuchsia, 350, 350, 350, 400)
        'window
        canvas.DrawRectangle(Pens.DarkTurquoise, 325, 250, 50, 50)

        'oval window
        canvas.DrawEllipse(Pens.Blue, 275, 125, 50, 20)

        'concentric circles
        canvas.DrawEllipse(Pens.Black, 50, 100, 100, 100)
        canvas.DrawEllipse(Pens.Black, 75, 125, 50, 50)

        'polygon
        pointlist(0) = New Point(200, 100)
        pointlist(1) = New Point(260, 120)
        pointlist(2) = New Point(300, 155)
        pointlist(3) = New Point(210, 210)
        pointlist(4) = New Point(140, 120)
        'canvas.FillPolygon(Brushes.Black, pointlist)

        'lake with gradient color
        lakebrush = New LinearGradientBrush _
        (New Rectangle(400, 300, 350, 100), Color.Aqua, Color.PaleVioletRed,
LinearGradientMode.Horizontal)

        canvas.FillEllipse(lakebrush, 400, 300, 350, 100)

        'draw clouds
        mycolorbrush = New SolidBrush(Color.FromArgb(200, 200, 200))
        canvas.FillEllipse(mycolorbrush, 5, 5, 50, 10)
        canvas.FillEllipse(mycolorbrush, 7, 10, 50, 10)

        rainpen = New Pen(Color.DarkBlue, 3)

        rainpen.DashStyle = DashStyle.DashDotDot

        canvas.DrawLine(rainpen, 5, 10, 300, 300)
    End Sub
End Class
```

Visual Basic Programming Task 2: Integrate a background image with a character created using Microsoft Paint or found on the Internet. Make the character respond to users' keyboard input, and provide some background music and sound effects.

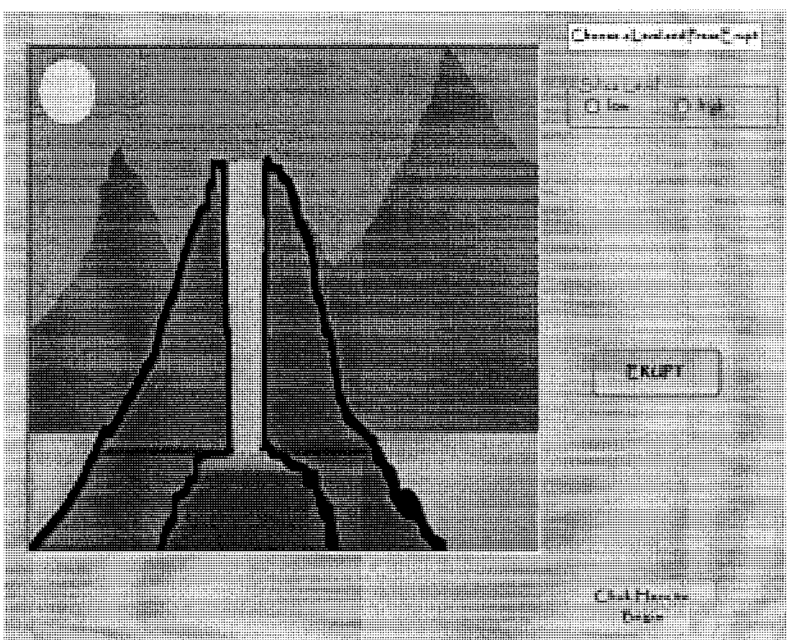



```
Imports System.Drawing.Drawing2D

Public Class Form1
   ' Declaration section - asks computers to create memory for this to increase
efficiency, also acts as a spell check for later on
   Private canvas As Graphics
   'create points for polygon
   Private waterpointList(5) As Point
   ' create custom brush

   Private oceanbrush As Brush

   Private fishingboat As Bitmap
   Private boot As Bitmap
   Private copepod As Bitmap
   Private menhaden As Bitmap
   Private shark As Bitmap
   Private giantsquid As Bitmap
```

```
    Private x As Integer
    Private y As Integer

    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click
        'start canvas
        canvas = picAnimation.CreateGraphics
        'load in pictures'
        fishingboat = New Bitmap("fishing boat.bmp")
        fishingboat.MakeTransparent(Color.Blue)
        boot = New Bitmap("boot.bmp")
        boot.MakeTransparent(Color.Blue)
        copepod = New Bitmap("copepod.bmp")
        copepod.MakeTransparent(Color.Blue)
        menhaden = New Bitmap("men.jpg")
        menhaden.MakeTransparent(Color.Blue)
        shark = New Bitmap("shark.bmp")
        shark.MakeTransparent(Color.Blue)
        giantsquid = New Bitmap("giant squid.bmp")
        giantsquid.MakeTransparent(Color.Blue)

        'brushes
        oceanbrush = New LinearGradientBrush(New Rectangle(0, 75,
picAnimation.Width, picAnimation.Height), Color.LightSkyBlue, Color.Black,
LinearGradientMode.Vertical)

        'sky
        canvas.FillRectangle(Brushes.LightBlue, 0, 0, picAnimation.Width, 200)

        ' ocean polygon
        waterpointList(0) = New Point(0, 150)
        waterpointList(1) = New Point(250, 125)
        waterpointList(2) = New Point(500, 150)

        waterpointList(3) = New Point(picAnimation.Width, 125)
        waterpointList(4) = New Point(picAnimation.Width, 500)
        waterpointList(5) = New Point(0, 500)

        canvas.FillPolygon(oceanbrush, waterpointList)

        'insert fishingboat
```

```
        canvas.DrawImage(fishingboat, 150, 70, 100, 100)
        'insert boot
        canvas.DrawImage(boot, 240, 160, 25, 25)
        'insert copepod
        canvas.DrawImage(copepod, 100, 300, 30, 30)
        'insert menhaden
        canvas.DrawImage(menhaden, 140, 300, 80, 50)
        'insert shark
        canvas.DrawImage(shark, 230, 300, 200, 150)
        'insert squid
        canvas.DrawImage(giantsquid, 440, 265, 220, 140)

        'objects in sky
        'sun
        canvas.FillEllipse(Brushes.Cornsilk, 550, 0, 100, 100)

    End Sub
End Class
```

Visual Basic Programming Task 3 (Final Project): Create an educational software or game that they could use to teach a lesson or principle in their school. It had to incorporate sound, an animated character, and support the user's keyboard or mouse input.



```
'------------------------------------------
'Purpose -  To Demonstrate how magma compostion affects volcanic eruptions
'------------------------------------------

Imports System.Drawing.Drawing2D
Imports System.Threading

Public Class Form1

    Private canvas As Graphics
    Private volcano As Bitmap
    Private magmabrush As Brush
    Private userquits As Boolean
    Private background As Bitmap
    Private audiobookmark As Double


    'sprite animation variables
    Private framewidth As Integer = 200
    Private frameheight As Integer = 200
    Private framex As Integer
    Private framerectangle As Rectangle
    Private spritescreenrectangle As Rectangle
    Private animatedmermaid As Bitmap
    Private animatedmagma As Bitmap

    'set up double buffer
    Private scratchcanvas As Graphics
    Private scratchpicture As Bitmap
    Private x As Decimal
    Private y As Decimal

    'quiet eruption animation variables
    Private framewidthquiet As Integer = 640
    Private frameheightquiet As Integer = 512
    Private framexquiet As Integer
    Private framerectanglequiet As Rectangle
    Private quietscreenrectangle As Rectangle
    Private animatedquieteruption As Bitmap
    Private endlowsilica As Boolean
    Private endhighsilica As Boolean
```

```
    Private endlowtemperature As Boolean
    Private endhightemperature As Boolean
    Private endlowgas As Boolean
    Private endhighgas As Boolean
    Private animatedviolenteruption As Bitmap
    Private looping As Integer

    'mermaid animation variables
    Private framewidthmermaid As Integer = 125
    Private frameheightmermaid As Integer = 125
    Private framexmermaid As Integer
    Private framerectanglemermaid As Rectangle
    Private screenrectanglemermaid As Rectangle

    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click

        Timer1.Start()
        canvas = picAnimation.CreateGraphics
        'start scratch canvas - sets up double buffer to eliminate flashing
        scratchpicture = New Bitmap(picAnimation.Width, picAnimation.Height)
        scratchcanvas = Graphics.FromImage(scratchpicture)
        background = New Bitmap("backgrounddrawn.bmp")

        'add lava bubbling sound to first screen
        mediaplayer1.URL = "quieteruption.wav"
        mediaplayer1.Ctlcontrols.play()
        'repeat sound a number of times
        mediaplayer1.settings.playCount = 1000
        'increase volume
        mediaplayer1.settings.volume = 100

        'add animated magma
        animatedmagma = New Bitmap("magma.bmp")
        ''make lava gradient
        'magmabrush = New LinearGradientBrush(New Rectangle(50, 50, 50, 100),
Color.Maroon, Color.OrangeRed, LinearGradientMode.Vertical)

        'canvas.FillRectangle(magmabrush, 100, 350, 200, 200)

        'add volcano picture
```

```
    volcano = New Bitmap("volcanofilled.bmp")
    volcano.MakeTransparent(Color.Blue)
    canvas.DrawImage(volcano, 0, 0, picAnimation.Width, picAnimation.Height)

    'load in animated mermaid
    animatedmermaid = New Bitmap("mermaidcopy.bmp")
    animatedmermaid.MakeTransparent(Color.Blue)

    'load in animated quiet eruption
    animatedquieteruption = New Bitmap("quieteruptionsequence.bmp")
    animatedquieteruption.MakeTransparent(Color.Blue)

    'load in animated violent eruption
    animatedviolenteruption = New Bitmap("violenteruptionsequence.bmp")
    animatedviolenteruption.MakeTransparent(Color.Blue)
    x = 100
    y = 350

    btnStart.Enabled = False

    Do

        TextBox1.Visible = True
        btnerupt.Visible = True

        scratchcanvas.DrawImage(background, 0, 0, picAnimation.Width,
picAnimation.Height)
        'scratchcanvas.FillRectangle(magmabrush, 100, 350, 200, 200)
        scratchcanvas.DrawImage(volcano, 0, 0, picAnimation.Width,
picAnimation.Height)

        ''grab current frame
        'framerectangle = New Rectangle(framex, 0, framewidth, frameheight)
        'spritescreenrectangle = New Rectangle(0.5 * x, 0.5 * y, framewidth,
frameheight)
        'scratchcanvas.DrawImage(animatedmermaid, spritescreenrectangle,
framerectangle, GraphicsUnit.Pixel)

        'grab current frame for magma
        framerectangle = New Rectangle(framex, 0, framewidth, frameheight)
        spritescreenrectangle = New Rectangle(x, y, framewidth, frameheight)
```

```
        scratchcanvas.DrawImage(animatedmagma, spritescreenrectangle,
framerectangle, GraphicsUnit.Pixel)

        'scratchcanvas.FillRectangle(magmabrush, 100, 350, 200, 200)
        scratchcanvas.DrawImage(volcano, 0, 0, picAnimation.Width,
picAnimation.Height)
        canvas.DrawImage(scratchpicture, 0, 0)
        Thread.Sleep(60)

        'framex = framex + framewidth
        'If framex >= animatedmermaid.Width Then framex = 0

        framex = framex + framewidth
        If framex >= animatedmagma.Width Then framex = 0
        Application.DoEvents()

      Loop Until userquits

      btnStart.Enabled = True
    End Sub

    Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
      userquits = True

    End Sub

    Private Sub btnerupt_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnerupt.Click

      'makes original sound stop when erupt button is pushed
      'mediaplayer1.Ctlcontrols.stop()


      'Change silica amount then silica box pops up and after erupt temperature box pops
up
      If GroupBox1.Visible = True Then


        If lowsilica.Checked = True Then
```

```
'quiet eruption animation sequence

Timendlowsilica.Start()

audiobookmark = mediaplayer1.Ctlcontrols.currentPosition
mediaplayer1.Ctlcontrols.stop()

'add quiet eruption noise
mediaplayer1.URL = "nonexpolsiveeruption.wav"
mediaplayer1.Ctlcontrols.play()
'repeat sound a number of times
mediaplayer1.settings.playCount = 1000
'increase volume
mediaplayer1.settings.volume = 100

Do

    scratchcanvas.DrawImage(background, 0, 0, picAnimation.Width,
picAnimation.Height)

    'grab current frame for quieteruption
    framerectanglequiet = New Rectangle(framexquiet, 0, framewidthquiet,
frameheightquiet)
    quietscreenrectangle = New Rectangle(0, 0, framewidthquiet,
frameheightquiet)
    scratchcanvas.DrawImage(animatedquieteruption, quietscreenrectangle,
framerectanglequiet, GraphicsUnit.Pixel)

    'scratchcanvas.FillRectangle(magmabrush, 100, 350, 200, 200)
    'scratchcanvas.DrawImage(volcano, 0, 0, picAnimation.Width,
picAnimation.Height)

    canvas.DrawImage(scratchpicture, 0, 0)
    Thread.Sleep(60)
    framexquiet = framexquiet + framewidthquiet
    If framexquiet >= animatedquieteruption.Width Then framexquiet = 0
    Application.DoEvents()

Loop Until endlowsilica 'userquits

mediaplayer1.URL = "quieteruption.wav"
```

```
        mediaplayer1.Ctlcontrols.currentPosition = audiobookmark
        mediaplayer1.Ctlcontrols.play()
        GroupBox1.Visible = False
        GroupBox2.Visible = True
        lowsilica.Checked = False
        Timendlowsilica.Start()

    End If

    If highsilica.Checked = True Then

        'violent eruption animation sequence
        Timeendhighsilica.Start()

        audiobookmark = mediaplayer1.Ctlcontrols.currentPosition
        mediaplayer1.Ctlcontrols.stop()

        'add violent eruption noise
        mediaplayer1.URL = "eruptingvolcano.wav"
        mediaplayer1.Ctlcontrols.play()
        'repeat sound a number of times
        mediaplayer1.settings.playCount = 1000
        'increase volume
        mediaplayer1.settings.volume = 100

        Do
            scratchcanvas.DrawImage(background, 0, 0, picAnimation.Width,
picAnimation.Height)

            'grab current frame for quieteruption
            framerectanglequiet = New Rectangle(framexquiet, 0, framewidthquiet,
frameheightquiet)
            quietscreenrectangle = New Rectangle(0, 0, framewidthquiet,
frameheightquiet)
            scratchcanvas.DrawImage(animatedviolenteruption, quietscreenrectangle,
framerectanglequiet, GraphicsUnit.Pixel)

            'scratchcanvas.FillRectangle(magmabrush, 100, 350, 200, 200)
            'scratchcanvas.DrawImage(volcano, 0, 0, picAnimation.Width,
picAnimation.Height)
```

```
            canvas.DrawImage(scratchpicture, 0, 0)
            Thread.Sleep(60)

            framexquiet = framexquiet + framewidthquiet
            If framexquiet >= animatedquieteruption.Width Then framexquiet = 0

            Application.DoEvents()


        Loop Until endhighsilica 'userquits

        mediaplayer1.URL = "quieteruption.wav"
        mediaplayer1.Ctlcontrols.currentPosition = audiobookmark
        mediaplayer1.Ctlcontrols.play()
        GroupBox1.Visible = False
        GroupBox2.Visible = True
    End If
End If

'change temperature and watch eruption
If GroupBox2.Visible = True Then
    If lowtemp.Checked = True Then

        'violent eruption sequence
        timeendlowtemperature.Start()
        audiobookmark = mediaplayer1.Ctlcontrols.currentPosition
        mediaplayer1.Ctlcontrols.stop()

        'add violent eruption noise
        mediaplayer1.URL = "eruptingvolcano.wav"
        mediaplayer1.Ctlcontrols.play()
        'repeat sound a number of times
        mediaplayer1.settings.playCount = 1000
        'increase volume
        mediaplayer1.settings.volume = 100

        Do
            scratchcanvas.DrawImage(background, 0, 0, picAnimation.Width,
picAnimation.Height)

            'grab current frame for quieteruption
```

```
                framerectanglequiet = New Rectangle(framexquiet, 0, framewidthquiet,
frameheightquiet)
                quietscreenrectangle = New Rectangle(0, 0, framewidthquiet,
frameheightquiet)
                scratchcanvas.DrawImage(animatedviolenteruption, quietscreenrectangle,
framerectanglequiet, GraphicsUnit.Pixel)

                'scratchcanvas.FillRectangle(magmabrush, 100, 350, 200, 200)
                'scratchcanvas.DrawImage(volcano, 0, 0, picAnimation.Width,
picAnimation.Height)

                canvas.DrawImage(scratchpicture, 0, 0)
                Thread.Sleep(60)
                framexquiet = framexquiet + framewidthquiet
                If framexquiet >= animatedquieteruption.Width Then framexquiet = 0

                Application.DoEvents()

            Loop Until endlowtemperature 'userquits

            mediaplayer1.URL = "quieteruption.wav"
            mediaplayer1.Ctlcontrols.currentPosition = audiobookmark
            mediaplayer1.Ctlcontrols.play()

            GroupBox2.Visible = False
            GroupBox3.Visible = True

        End If
        If hightemp.Checked = True Then

            'non violent eruption sequence

            timeendhightemperature.Start()

            audiobookmark = mediaplayer1.Ctlcontrols.currentPosition
            mediaplayer1.Ctlcontrols.stop()

            'add quiet eruption noise
            mediaplayer1.URL = "nonexpolsiveeruption.wav"
            mediaplayer1.Ctlcontrols.play()
            'repeat sound a number of times
```

```
            mediaplayer1.settings.playCount = 1000
            'increase volume
            mediaplayer1.settings.volume = 100

            Do
              scratchcanvas.DrawImage(background, 0, 0, picAnimation.Width,
picAnimation.Height)

              'grab current frame for quieteruption
              framerectanglequiet = New Rectangle(framexquiet, 0, framewidthquiet,
frameheightquiet)
              quietscreenrectangle = New Rectangle(0, 0, framewidthquiet,
frameheightquiet)
              scratchcanvas.DrawImage(animatedquieteruption, quietscreenrectangle,
framerectanglequiet, GraphicsUnit.Pixel)

              'scratchcanvas.FillRectangle(magmabrush, 100, 350, 200, 200)
              'scratchcanvas.DrawImage(volcano, 0, 0, picAnimation.Width,
picAnimation.Height)

              canvas.DrawImage(scratchpicture, 0, 0)
              Thread.Sleep(60)

              framexquiet = framexquiet + framewidthquiet
              If framexquiet >= animatedquieteruption.Width Then framexquiet = 0

              Application.DoEvents()

            Loop Until endhightemperature 'userquits

            mediaplayer1.URL = "quieteruption.wav"
            mediaplayer1.Ctlcontrols.currentPosition = audiobookmark
            mediaplayer1.Ctlcontrols.play()

            GroupBox2.Visible = False
            GroupBox3.Visible = True

        End If
    End If

    'change gas content and watch eruption
```

```
If GroupBox3.Visible = True Then
   If lowgas.Checked = True Then

      'quiet eruption sequence

      timeendlowgas.Start()

      audiobookmark = mediaplayer1.Ctlcontrols.currentPosition
      mediaplayer1.Ctlcontrols.stop()

      'add quiet eruption noise
      mediaplayer1.URL = "nonexpolsiveeruption.wav"
      mediaplayer1.Ctlcontrols.play()
      'repeat sound a number of times
      mediaplayer1.settings.playCount = 1000
      'increase volume
      mediaplayer1.settings.volume = 100

      Do
         scratchcanvas.DrawImage(background, 0, 0, picAnimation.Width,
picAnimation.Height)



         'grab current frame for quieteruption
         framerectanglequiet = New Rectangle(framexquiet, 0, framewidthquiet,
frameheightquiet)
         quietscreenrectangle = New Rectangle(0, 0, framewidthquiet,
frameheightquiet)
         scratchcanvas.DrawImage(animatedquieteruption, quietscreenrectangle,
framerectanglequiet, GraphicsUnit.Pixel)

         'scratchcanvas.FillRectangle(magmabrush, 100, 350, 200, 200)
         'scratchcanvas.DrawImage(volcano, 0, 0, picAnimation.Width,
picAnimation.Height)

         canvas.DrawImage(scratchpicture, 0, 0)
         Thread.Sleep(60)

         framexquiet = framexquiet + framewidthquiet
         If framexquiet >= animatedquieteruption.Width Then framexquiet = 0
```

```
        Application.DoEvents()

    Loop Until endlowgas 'userquits

    mediaplayer1.URL = "quieteruption.wav"
    mediaplayer1.Ctlcontrols.currentPosition = audiobookmark
    mediaplayer1.Ctlcontrols.play()

    GroupBox3.Visible = False

    btnrestart.Visible = True

    GroupBox3.Visible = False
    GroupBox1.Visible = False
    lowsilica.Checked = False
    highsilica.Checked = False
    lowtemp.Checked = False
    hightemp.Checked = False
    lowgas.Checked = False
    highgas.Checked = False

End If

If highgas.Checked = True Then
    'violent eruption sequence
    timeendhighgas.Start()

    audiobookmark = mediaplayer1.Ctlcontrols.currentPosition
    mediaplayer1.Ctlcontrols.stop()
    'add violent eruption noise
    mediaplayer1.URL = "eruptingvolcano.wav"
    mediaplayer1.Ctlcontrols.play()
    'repeat sound a number of times
    mediaplayer1.settings.playCount = 1000
    'increase volume
    mediaplayer1.settings.volume = 100

    mediaplayer2.URL = "scream.wav"
    mediaplayer2.Ctlcontrols.play()
    'repeat sound a number of times
```

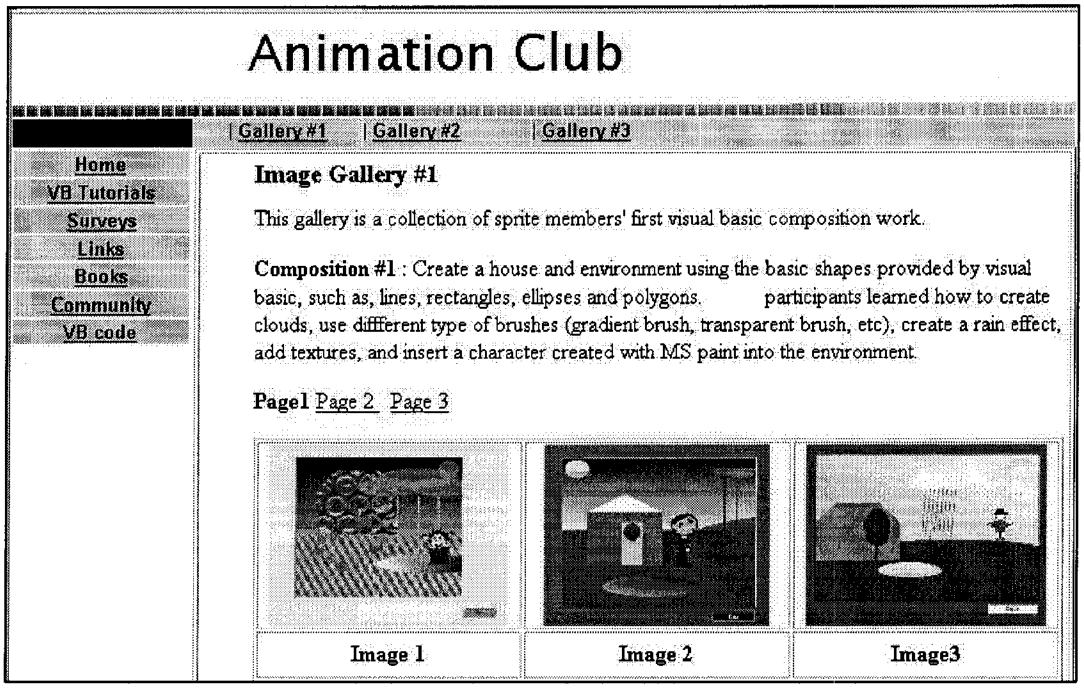
```
        mediaplayer1.settings.playCount = 1
        'increase volume
        mediaplayer1.settings.volume = 100

        Do
            scratchcanvas.DrawImage(background, 0, 0, picAnimation.Width,
picAnimation.Height)

            'grab current frame for quieteruption
            framerectanglequiet = New Rectangle(framexquiet, 0, framewidthquiet,
frameheightquiet)
            quietscreenrectangle = New Rectangle(0, 0, framewidthquiet,
frameheightquiet)
            scratchcanvas.DrawImage(animatedviolenteruption, quietscreenrectangle,
framerectanglequiet, GraphicsUnit.Pixel)

            'scratchcanvas.FillRectangle(magmabrush, 100, 350, 200, 200)
            'scratchcanvas.DrawImage(volcano, 0, 0, picAnimation.Width,
picAnimation.Height)

            'grab current frame
            framerectanglemermaid = New Rectangle(framexmermaid, 0,
framewidthmermaid, frameheightmermaid)
            screenrectanglemermaid = New Rectangle(200, 50, 0.5 * framewidth, 0.5 *
frameheight)
            scratchcanvas.DrawImage(animatedmermaid, screenrectanglemermaid,
framerectanglemermaid, GraphicsUnit.Pixel)

            canvas.DrawImage(scratchpicture, 0, 0)
            Thread.Sleep(60)

            framexquiet = framexquiet + framewidthquiet
            If framexquiet >= animatedquieteruption.Width Then framexquiet = 0

            framexmermaid = framexmermaid + framewidthmermaid
            If framexmermaid >= animatedmermaid.Width Then framexmermaid = 0
           Application.DoEvents()

        Loop Until endhighgas 'userquits

        mediaplayer1.URL = "quieteruption.wav"
```

```
            mediaplayer1.Ctlcontrols.currentPosition = audiobookmark
            mediaplayer1.Ctlcontrols.play()

            GroupBox3.Visible = False

            btnrestart.Visible = True
            lowsilica.Checked = False
            highsilica.Checked = False
            lowtemp.Checked = False
            hightemp.Checked = False
            lowgas.Checked = False
            highgas.Checked = False

        End If
    End If
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    GroupBox1.Visible = True
    Timer1.Stop()
End Sub

Private Sub Timendlowsilica_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timendlowsilica.Tick
    endlowsilica = True
    Timendlowsilica.Stop()
End Sub
Private Sub Timeendhighsilica_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timeendhighsilica.Tick
    endhighsilica = True
    Timeendhighsilica.Stop()
End Sub

Private Sub timeendlowtemperature_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles timeendlowtemperature.Tick
    endlowtemperature = True
    timeendlowtemperature.Stop()
End Sub
Private Sub timeendhightemperature_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles timeendhightemperature.Tick
```

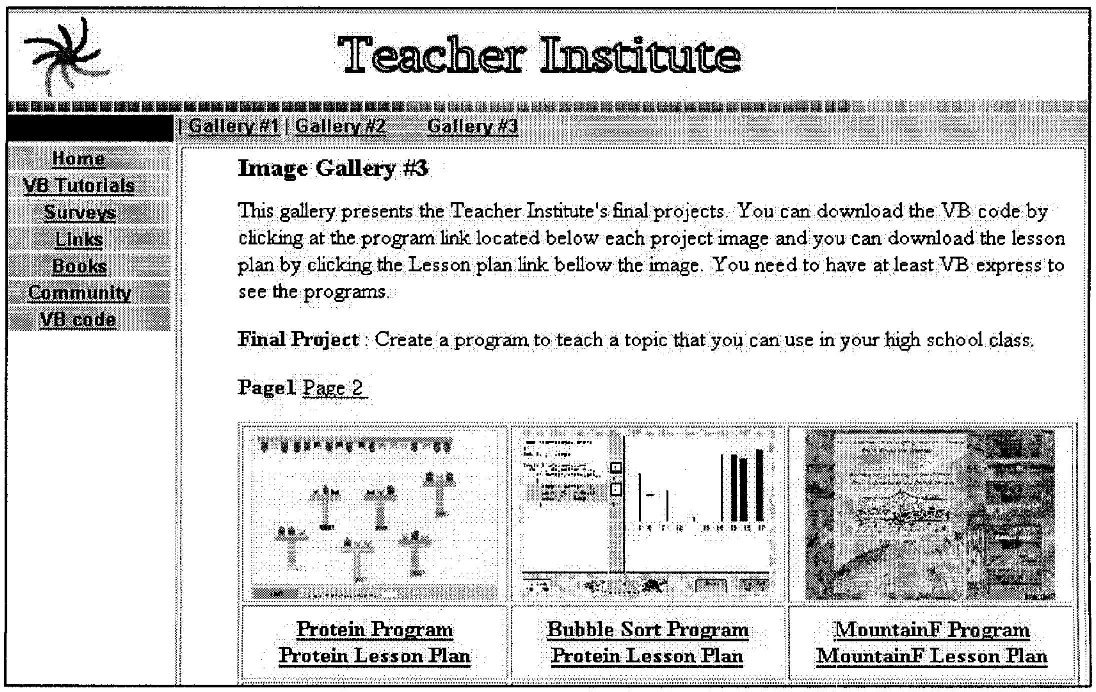
```
        endhightemperature = True
        timeendhightemperature.Stop()
    End Sub

    Private Sub timeendlowgas_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles timeendlowgas.Tick
        endlowgas = True
        timeendlowgas.Stop()
    End Sub

    Private Sub timeendhighgas_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles timeendhighgas.Tick
        endhighgas = True
        timeendhighgas.Stop()
    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    End Sub
    Private Sub lowsilica_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles lowsilica.CheckedChanged
        If lowsilica.Checked Then
        End If
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnrestart.Click
        Application.Restart()
    End Sub
End Class
```

Appendix E: Class Websites and Online Communities Screenshots

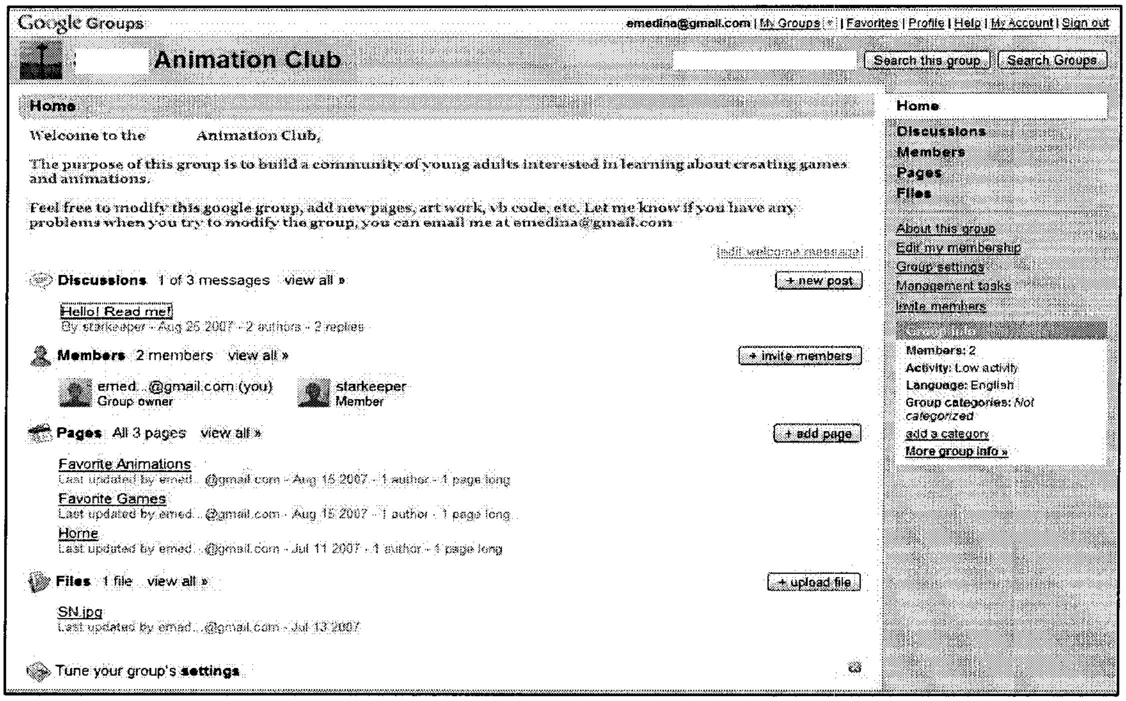
Students' Class Website Screenshot



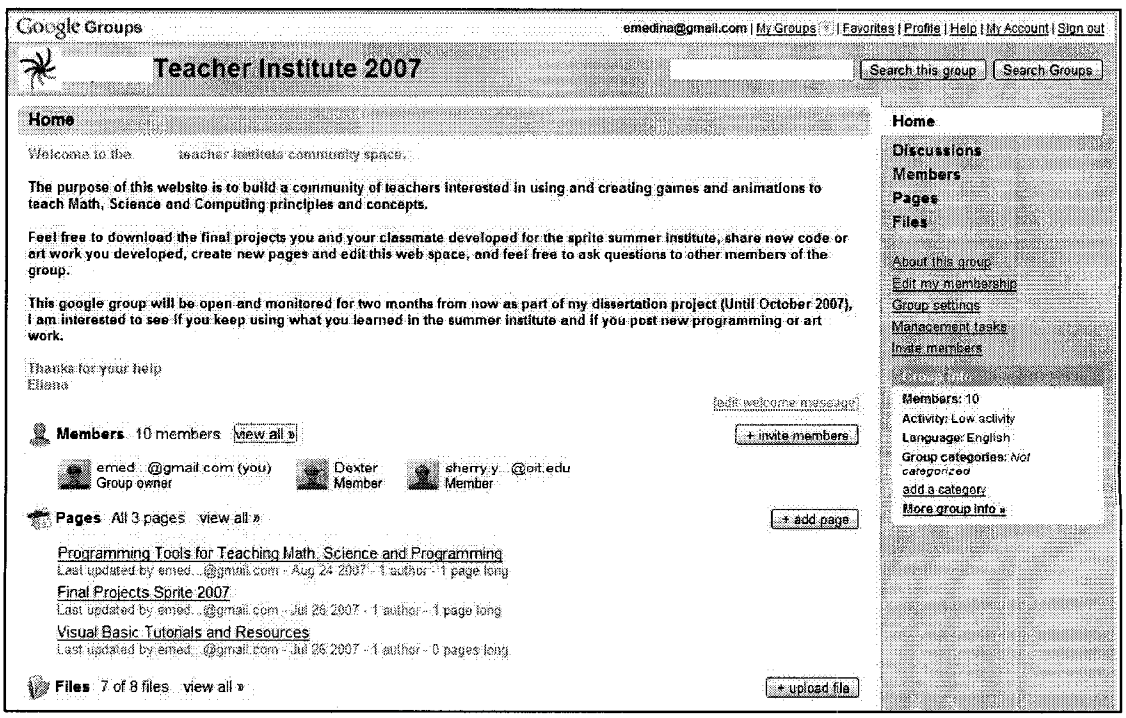
Teachers' Class Website Screenshot



Students' Online Community Screenshot



Teachers' Online Community Screenshot



Appendix F: Summer Camp and Teacher Institute Learning Activities

Summer Camp Activities

	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1	- Students get material (badge, t-shirt, thumb drive, notebook, and pen). - Students create passwords for their computers -Lecture about what animation is, who is involved in creating animations, cost of animation. - Students take test about mathematic knowledge. -Showcase of previous year students' final project. -Ice breaker game.	-Lecture about use of animation in science. -Getting started with VB (creating scenes using geometric figures). -Elements of Story Telling. -More VB (using brushes, textures, and creating characters in Paint). -First Photoshop session (create two characters). - Students receive consent Forms and letter of invitation to the research study.	No class, 4th of July.	-Review of VB. -Continue working on the scene in VB (create clouds using ellipses and transparent brushes). - Integrate character into new VB project. - Photoshop session (create a background image).	- Review. - Group exercise about matrix multiplication. -Use VB to create a background scrolling effect. -Show different ways to use matrix rotation in VB to create a particular animated effect. Showed previous year projects. -Contest about how to do matrix multiplications. -Animation techniques to create smooth animations in VB (double buffering). -Photoshop class (Clay session).
Week 2	- Review of the matrix multiplication and flickering problem. - Adding sounds and music to the program. - Mathematic test. - Form groups for final project. (volunteer basis) - Add stories and using timers.	- Review - Continue working on VB sound and adding story. - Use timers and interactions to change the storyline. -Work on research project. - Photoshop session.	- Review. -Animating a sprite. -Students' presentation of research project (using animation in a particular science field). -Students work on their research presentation. -Introduction to modular arithmetic to slow down the speed of the animation. -Students	-Review matrix multiplications. - Animation ideas using 5 drawings (how to make a candle look like is melting) -More explanation about timers. -Research Project presentations. - Students work on final project	- Review of timers - Introduction to physics in animation (Newton laws, parabola, line, inverse parabola). -Adding physics to the VB code to implement jumps, increase animation speed gradually. - Finish research project presentations. - TAs' presentation about salaries in

			learn to put together the clay models in Photoshop to animate them.		IT, getting college credits in high school, what CS courses look like in college, how math is used to create games. -Work on final project
Last Day (Saturday)	-Review of using physics for games - New content about how to use advance math and physics equations to create games. - Test about math and physics. - Work on project. -Final Project Presentations.				

Summer Institute Activities

	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1	- Teachers get material (badge, thumb drive, notebook, and pen). - Introduction to the summer institute. - Lecture about underrepresentation of women and minority students in Computer Science education. - Lecture about why games and animations are good to motivate students. -Talk about Students' summer camp and showcase of students' final project. -Lecture about learning styles. - Ice breaker game. - Teachers create passwords for their computers - Getting started with VB.	-Lecture about visual basic. -Using VB to create a scene using geometric figures. - Showcase of previous year teachers' final project. - Teachers receive consent Forms and letter of invitation to the research study.	- Review of VB. - Continue working on the scene in VB (create clouds using ellipses and transparent brushes). - Integrate character into new VB project. - Showcase of students' final project.	-Review of VB. -Continue working on VB. - Instructions about the final project - Start working on final projects in their groups. - Lecture about using the keyboard input in VB to add interactions in a game.	- Review. - Lecture about matrix operations. -Animation techniques to create smooth animations in VB (double buffering). - Showcase of previous year teachers' final project. - Learn about clay animation. Create clay models. - Review of matrix operations.
Week 2	- Review. - Adding sounds and music to the program. - Creating background images. - Use timers to change background image. - Teachers work on their final project.	- Teachers work on their final project.	- Teachers work on their final project.	- Teachers work on their final project. - Some teachers present their final project to the class.	- Teachers work on their final project. - Final Project Presentations.

Vita

ELIANA C. MEDINA

Education

- Ph.D. Cognitive Studies, Educational Technologies, University of Washington, Seattle USA, 2008
- B.S. Computer Science, Central University of Venezuela, Caracas VZ, 2000

Summary of Qualifications

- Three years research experience in human factors, educational technologies, games in education, and 3D visualization tools (planning, designing, researching and reporting of findings).

- Experience evaluating, creating and customizing different types of software applications, including learning management systems, games, graphic file format converters, database management systems, industrial plant designs and educational software.

- Solid understanding of computer programming, animation, game development, and emerging technologies.

- Educational and Usability Research as it applies to gaming and education, virtual reality, and new markets.

- Technical skills including fluency in C/C++, MsSQL, Access, VB, SPSS, Maple, Python, HTML, and Microsoft Office.

- Fluent in both English and Spanish.

- Excellent interpersonal and team skills. Works well independently and in team projects.

Professional / Academic Experience

- **10/06 – Present Ackerley Partnership Project, UW College of Education, Seattle WA, Research Assistant:** Create and maintain a database of sixteen-teacher professional development projects in the Puget Sound area. Design and conduct research on college students using a system to annotate videos called "Video Traces".

- **1/05 – 6/07 UW Game and Simulation Group, Seattle WA, Co-Founder:** Focus Group researching and developing computer games for educational purposes.

- **11/05 – 11/06 UW Making Connections Mentoring Program, Mentor:** Mentoring for High School girls to pursue higher education, engineering, and game development.

- **9/03 – 6/06 Human Interface Technology Lab (HITL), Seattle WA, Research Assistant:** Conducted human factors research on individual differences in adaptation to walk inside a device that simulates normal walking, called "the VirtuSphere", to navigate virtual reality worlds and virtual video games. Designed, collected data, analyzed the data, and wrote the report. Worked with an interdisciplinary team from the UW-Human Interface Technology Lab and SCRIPPS Research Institute to create and evaluate emerging technologies that can help biochemistry students to get a better understanding of protein structure. This project was called "The Protein Magic book" and "The Amino Acid Augmented Reality".

- **5/03 – 9/03 Luz de Caracas Elementary School / Venezuelan-American Center, Caracas VZ:** ESL teacher Taught English in an elementary school (kindergarten through sixth grade) and adult learning English institution.

- **4/02 – 12/02 DKE, Inc, Seattle WA, Programmer and Researcher**: Designed and developed Database applications using Access & Visual Basic. Researched about the usability of E-learning software for an airline industry engagement.

- **4/98 – 12/01 Laboratory of Computer Graphics and Applied Geometry, Caracas VZ, Programmer:** Created solutions to study and display custom graphic surfaces to model real life objects. Saved approximately two months of programming time using customized math scripts using Maple instead of using C++ libraries. Web Development including designing and programming a department based HTML portal which included project information from all previous lab projects, collecting and incorporating research project outcomes. Taught courses for the Plant Design System though the Dept. of Mathematics which taught engineers how to use databases and script programming technologies to develop complex industrial plants as part of a training cooperation program with Intergraph, Inc.

Conferences & Papers

- **2008 Human Factors and Ergonomics Society (HFES), NYC NY:** Papers accepted for presentation on Virtusphere and Eye tracking systems.
- **2008 Technical Symposium on Computer Science Education (SIGCSE), Portland OR:** Presentation at the doctoral consortium.
- **2007 World Conference on Educational Multimedia, Hypermedia, and Telecommunications (EDMEDIA), Vancouver BC:** Paper presented for Proceedings.
- **2006 Microsoft Faculty Summit, Redmond WA:** Representative for Human Interface Technology Lab, UW
- **2006 CGA Casual Games Connect, Seattle WA:** Volunteer
- **2006 American Educational Research Association Conference (AERA), San Francisco CA:** Paper presented as part of a round table on motivating Hispanic American Girls through computer Games.
- **2005 Digital Games Research Association Conference (DIGRA), Vancouver BC:** Paper presented as part of a mentoring round table on motivation and digital games.
- **2005 Society for Technical Communication Conference, Seattle WA:** Panel presentation on engagement and localization in digital games.
- **2004 Game Developer Conference (GDC), San Jose CA:** IGDA Scholarship recipient.
- **2003 Venezuelan-American Center, Caracas VZ:** Lesson Planning Workshop to teach English 2003.
- **2002 Smartgirls Digital Imaging Workshop, Seattle WA:** Volunteer.
- **2002 Game Developer Conference (GDC), San Jose CA:** Volunteer.
- **1999 – 2002 ACM Special Interest Group on Computer Graphics (SIGGRAPH), New Orleans LA, Los Angeles CA, San Antonio TX:** Volunteer.

- **2001 Latin American Workshop on Internet Technology Networks (WALC), Mérida VZ:** Attended a training program to improve information services in Latin America.

- **2000 Association for the Advancement of Science (ASOVAC), Caracas VZ:** Presented about graphic format files for computerized medical images.