Gaming for Middle School Students: Building Virtual Worlds

Charles R. Hardnett
Spelman College
Computer Science Department
Atlanta, GA 30314
hardnett@spelman.edu

ABSTRACT
We can begin engaging new talented students Computer Science students with Gaming as the introduction. Students are already excited and curious about games. Research studies suggest students form opinions about whether they enjoy science or non-science subjects as early as middle school. Furthermore, a student entering high school makes course selections based on their interests of preparing for college or not. The college prep course sequences are the best sequences to prepare students for college science majors including Computer Science. Therefore, it is important for the Gaming and Computing industries to focus our attention and resources on building young talent at the middle school and early high-school years. The Computer Science Department at Spelman College has implemented one approach to address this need. We have implemented summer camp with follow-up workshops during the subsequent school year. Our summer camps are called CARE Summer Computer Camps and are funded by participant fees subsidized by an NSF [7] Broadening Participation in Computing (BPC) grant. One of the major components of the CARE Camps is to introduce students to computer programming by building interactive virtual worlds. Constructing interactive virtual worlds is a necessary component to building games. Students are introduced to the design and implementation aspects of game development through virtual world development. In this paper we present our CARE curriculum and the results of our evaluations. We demonstrate that students are more excited about Computing and Gaming after their camp experience and are interested in pursuing further studies in Gaming and Computing.

Categories and Subject Descriptors
I.3.8 [Computing Methodologies]: Computer Graphics—Applications; K.3 [Computer and Education]; K.8 [General]: Games

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

1. INTRODUCTION
The Gaming industry is concerned about having the talent required to build the next generation of gaming software and hardware. This concern is another symptom in a long list that can be attributed to the low enrollment numbers for Computer Science across the country. There are several reasons why there are low enrollments such as students not being prepared for the CS curriculum [9], students not interested in the CS curriculum [8], students not aware of the opportunities in computer caused inaccurate perceptions formed from the “.com bust” of the 90’s, students afraid that CS is too hard and they can not do it, etc [2].

We have begun to address this problem at our institution by pursuing various relationships with K-12 programs that range from organizing and implementing technology workshops to weekend and summer camps for K-12 students to developing Computing curriculum modules for K-12 classrooms. In this paper, we present the CARE Summer Computer Camp as a avenue for introducing students to programming via interactive 3-D worlds. These 3-D virtual worlds can be games or animated stories. The stories can be made interactive or multi-threaded. For example, allowing the user to control the sound effects or having the user interact with the story in a question/answer format that enables the user to alter the outcome of the story. We have chosen to use Alice [5] as the platform for our graphics programming in the CARE Computer Camps. Alice provides a very suitable platform to introduce students to various elementary aspects of building video games and animated stories. The interface is very accessible to a wide range of ages and abilities.

Our focus on middle school students is not by accident. Research shows that middle school ages is where students form valuable opinions about science and mathematics. This is the stage of life where students decide whether they are interested in mathematics and science or other non-science fields. This phenomenon is especially true for girls [10]. As a result, we decided that we should target middle school students and engage them with exciting and interesting Computing activities.
2. RELATED WORK

The CARE Camp motivation is clear from the previous section. There are other programs and camps that have inspired this work or offer more support for the importance of these camps. In this section, we describe some of these efforts. Calvin College and Lamar University have developed summer camps targeting female students [1] [6]. These camps were motivated by the negative experiences of female students in CS1 courses and the lack of female representation in computing. Their research determined that the negative experiences and negative feelings about computing were developed at very early ages. Female students associated computing with “geeks” and “nerds” prior to reaching high school. As a result, they were much less likely to consider computing careers and did not prepare for computing careers while selecting courses or developing hobbies while in high school. Therefore, when entering computer science programs in college they would feel ill-prepared. These camps were developed around the same time-frame as the CARE Camp, and support the rationale of targeting middle school students for these camps before negative perceptions are conceived. In addition, after school programs and other camps have started to use gaming as a center point to engage students in learning computing [12]. Werner et al developed an after school program for middle school students to develop IT knowledge. Their goals are to increase IT skills, knowledge of IT fundamental concepts, and intellectual capabilities. There goals are realized via gaming as the vehicle for discussion and projects. Other programs have also used gaming as the focus for teaching computing to a broad audience of middle school students [11]. These programs and others have similar motivations as the CARE Summer Camp. The CARE Summer Camp aims to target underrepresented boys and girls, and focuses on the design and implementation games and interactive virtual worlds. The next section of this paper we will describe this portion of the CARE Camp and its curriculum.

3. THE CARE CAMP METHODS

The CARE Summer Computer Camp is a one-week day camp. The camp is intended for middle school students that have completed the 6th, 7th, or 8th grades. While the CARE Camp is a computer and robotics camp, the computer graphics portion is featured in this paper. The graphics and animation curriculum are the following:

1. Teach students to a design process for developing computer animation and virtual worlds programs.
2. Provide instruction on how to use design tools such as storyboards.
3. Provide instruction on how to use a programming environment such as Alice to implement their designs.
4. Develop basic debugging and analytical skills for implementing computer animation and virtual worlds programs.

These goals are achieved through a variety of exercises and assignments during the week. These assignments introduce students to the virtual world development process and is discussed in the next section.

3.1 Virtual World Curricular Concepts

The curriculum for the graphics animation and virtual worlds development contains many aspects that are based on the principle of giving the students hands-on experiences to facilitate more efficient learning. In this section we discuss the various elements of our curriculum to support the curricular goals.

The curriculum begins with an introduction to the gaming industry and interactive media. We explore examples of games and discuss the difference between interactive and non-interactive virtual worlds. We also do a comparative analysis of games, simulations, and animated movies within this context. For example, the elements of a game in terms of animation and development of a virtual world are very similar to the elements used for computer animated movies. We point out the difference is that the player of a game can alter the outcome of the story. The player alters this outcome by controlling the characters in the story and making decisions for the characters. We also introduce them to serious games to provide perspective on the wide array of activities that have close connections with gaming technology. This introduction also includes an explanation of the types of software tools used by professionals in building console games and computer games. The last part of the introduction focuses on the software life-cycle applied to game design. We simplify the life-cycle to only include the following steps:

Step 1. Design This is the point of the process where you structure your ideas and develop documents and drawings to express your gaming concept.

Step 2. Implementation This step is for you to take your design documentation and create your idea using some collection of software for developing the graphics and the programming.

Step 3. Testing & Debugging This step is where you test your implementation to locate your errors and correct them. Correcting errors may require you to revisit the Design step or the Implementation step.

Step 4. Production This is the step where you deliver your game to your customers.

The introduction presents a technical, economic, and social overview of the gaming and animated movie industries. In addition, the students are introduced to the development process shared by both industries. The next section conveys how the students learn the development process in a hands-on fashion.

The design phase in our curriculum is centered around the development of storyboards using the template shown in Figure 1. Many of the students have never heard of a storyboard. Our storyboard template contains the components: scene number, sketch, description, and sound and text information. The scene number is used to organize the sequence of the storyboards. The sketch area provides a space to put a drawing or sketch of the scene. The description area provides the location to put detailed instructions for animation and interactivity for the scene. The sound and text information sections provide a place for user implicit and explicit feedback information for the scene. The first activity for the design phase is where students reverse-engineer the storyboards for an existing virtual world. Alice
has a wide assortment of example worlds. The students choose either an interactive or non-interactive virtual world. They create 3-4 storyboards that describe the chosen virtual world. This reverse-engineering activity allows students to build storyboards and focus on the mechanics behind them while ignoring the creative effort required. The next exercise allows them to utilize their creative talents in creating storyboards. In this activity, we provide the students with a set of scenarios. The students choose one of the scenarios to develop 3-4 storyboards. These scenarios loosely describe a children’s educational game, a video game, an animated story, or a simulation. The loose descriptions provide a natural opening for student creativity. After the students have learned to design their virtual worlds, we introduce them to the next phase of development using Alice [5] to implement their virtual world.

The students implement their virtual worlds using Alice. Alice is an interactive GUI programming environment for developing 3-D virtual worlds. A screen shot for Alice is shown in Figure 2. Alice was developed at CMU and has some very important features that support our goals:

1. Alice has a drag-n-drop interface for building programs using tiles that represent and resemble typed instructions.
2. Alice has a gallery of 3-D models of characters, environments, and props that can be used with any Alice virtual world.
3. Alice has incorporates fundamental principles of 3-D worlds along with CAD-like tools for manipulating the setting of the world.
4. Alice provides facilities for importing sound.
5. Alice provides facilities to enable keyboard and mouse interaction with the virtual world.

The first two features of Alice allows students to rapidly build their worlds. The gallery eliminates the need for students to learn 3-D modeling programs such as Maya [3] or Milkshape [4] to create 3-D models for their virtual worlds. Students browse the gallery for the objects they want to use in their virtual world, select the objects, and place the objects using a drag-n-drop interface. The drag-n-drop interface also eliminates syntax errors when building the programs, and thus decreases implementation time. In addition, using Alice introduces students to the use of the X,Y,Z Cartesian space in order to properly place and move characters in their stories. These are key reasons why we believe the Alice environment or an Alice-like environment to be ideal for beginning students at this level.

The implementation phase is where the majority of our instructional time is used during the week. We have the students perform several tutorials and exercises that expose
them to all of the necessary fundamentals in Alice to enable them to build interactive 3-D worlds. We begin by explaining the difference between sequential, conditional, and iterative operations. In the Alice interface shown in Figure 3 the program is shown as a collection of colored tiles. Instructions within control structures are shown as nested tiles. Control structures are special tiles at the bottom of the screen and include: loop, if-then, do together, while, and comment. The sequential instructions are method calls that are matched with each object in their world; the tiles for the methods are found on the left-hand side of the Alice screen in Figure 2. In our camp, the students learn the use of these tiles by building an example world with the instructor leading. This example world incorporates simple uses of these three types of operations. The students are asked to make small modifications to this virtual world that reinforces their understanding of the statements. In addition to these three categories, students learn about concurrency when we discuss ways to have the appendages of characters move at the same time to simulate walking, flying, running, throwing, etc. Students are always given a world to start with, and then perform necessary modifications to the world’s setup and the programming to reach the desired solutions. The next area of discussion involves event handling. Event handling in Alice is used to monitor and react to keyboard events and mouse events. When the player presses a key on the keyboard or moves the mouse an event is generated. These events have to be captured and processed by methods that are built by the students. Finally, we teach them how to use prerecorded sounds to add music and sound effects to their virtual worlds.

At the completion of the implementation instruction, students are prepared to take on a 2 day project that allows them to create their own worlds from concept to implementation.

3.2 Major Project

In this section, the major project for the week is presented. The major project is given to the students an opportunity to synthesize the various skills they have been introduced to during the first three days. The project is presented to the students where they are to create an interactive or non-interactive virtual world. The students that consider making an interactive virtual world are given further instructions that they may conceive of their interactive virtual world as a game or an interactive story. An interactive story is defined as a traditionally non-interactive virtual world like a short-story that provides mechanisms that allow the user to interact with the story. Some examples include developing a multi-threaded story where the user is asked questions about the direction they would like to see the story take; or having special keys that invoke sound effects during the story or short animation sequences. To get started students are urged to consider examining Alice textbooks and resources, commercials, movies, fictional books, and personal experiences as inspirations for their virtual worlds. At the completion of their projects, the students are asked to present their projects to the entire group of students and parents during a special open-house celebration.

In this section, the goals and curriculum are presented for the one-week CARE Camp for middle school students. The curriculum is designed to give students hands-on exposure to the development of virtual worlds to be used in games and in animated stories. The curriculum is filled with exercises that provide an opportunity for students to learn design and implementation principles. In addition, each student is expected to complete a project from concept to implementation in a team with one other student or as an individual. The concluding activity synthesizes all of the skills learned during the week into a single project that is then presented to other students and parents.

4. RESULTS AND DISCUSSION

The CARE Summer Computer Camps and workshops have been evaluated from December 2005 - August 2007. During this time we have had over 150 students participate in Summer Camps and workshops. In addition, 99% of the students have been African-American; 56% of the participants have been female and 44% have been males. For each camp and workshop we have opening and closing surveys taken by each student. In this section, the results of these surveys is discussed to show the effectiveness of the CARE program.

The students are asked questions about their overall experience with the Workshops and Camps. As seen in Figure 4. The students are overwhelmingly excited about the CARE programs and their experiences. The significance here is that many of the students expressed reservations about attending the CARE programs in the free response section of the pre-survey. However, after attending the programs the students are excited about the experience and they value the experience. We also asked the students questions about their opinions about the educational value of attending the CARE programs. Students responded at a 99% rate that they “agree a lot” that the CARE programs were educational for them.

Since nearly none of the students have ever had the experience of programming a computer, we wanted to determine their impressions about programming 3-D graphics and animation using Alice. The surveys showed that 90% of the students thought that programming in Alice was fun and exciting. This sentiment was coupled with 65% agreeing that 3-D graphics and animation programming was a challenge. These results show that CARE programs were able to strike a balance between the level of fun and level of difficulty. This is a significant result in our eyes, and gives us encouragement for developing future programs for middle and high school students that may be more challenging. It also shows
that even when the work is difficult and challenging, the students still have fun doing it and are willing to learn the skills required.

One of our goals of the CARE Programs is to attract students to the gaming and computing areas. We decided to measure this impact by determining how many students would be interested in learning more as they go into high school. Students that are willing to do this can remain in the pipeline, and we as a community will have opportunities to further hone their skills and interests. Our surveys show that 79% percent of the students were inspired to learn more about computing in high school. We also found that 3.5% more students were inspired enough by the one week camp to consider a career in computing. We realize that this is not as useful due to the ages of the participants; however, it does show that the students realize they can make a career in computing and almost 4% more of them are now turning in that direction.

We presented our data showing that students enjoy programming virtual worlds. We have also shown that programs such as the CARE program can possibly have positive impact on the students’ academic choices in high school to prepare for computing careers.

5. CONCLUSIONS AND FUTURE WORK

The overall goals of the CARE Summer Computer Camps and Workshops is to bring computing to middle school students to make them aware of the exciting nature of computing and the potential opportunities in their future. We feel that continued exposure to computing at younger ages can make an impact on the number of students entering computing areas in undergraduate and graduate school because they will be excited about the opportunities and prepare properly in high-school to succeed. Virtual world creation is a fascinating way to introduce students to Computing. Students come in contact with virtual worlds through animated movies and video games, and are often fascinated about the prospect of creating their own worlds that which they can control. This is also one of the starting points for introducing students to elementary aspects of game design. We chose to use Alice as our platform, but there are many other potential platforms. The contribution our camps have to the students is introducing them to virtual world design and game design as well as algorithmic problem solving, computer programming, computer graphics, image manipulation, and sound manipulation. The results from our surveys show that students are excited about using and learning the gaming industry concepts and technologies. In addition, our surveys suggest that camps and programs like the CARE program can have effects on how students view computing as a career option and how they may matriculate through high-school as they prepare for college. Our future directions are to mobilize the CARE programs to reach more students via a mobile version of the CARE Camps and Workshops. In addition, we are interested in exploring the use of Phrogram and GameMaker as potential tools for older students.

6. REFERENCES


