GAME DESIGN AND DEVELOPMENT COURSE TAUGHT WITH ALICE

Karen Villaverde, Daniel Jaramillo
New Mexico State University
1290 Frenger Mall SH 123, Las Cruces, NM 88003
575-646-3723
kvillave@cs.nmsu.edu, djaramil@cs.nmsu.edu

ABSTRACT
In this paper we describe our very positive experience in teaching a game design and development course with Alice as a 3D game development platform. We describe why Alice was chosen as our 3D game development platform, what material was covered, how the course was conducted, the quality of the students’ game projects, features of Alice that students liked for game development, problems that students had when using Alice, and the evaluation of the course.

1. INTRODUCTION
We always wanted to teach a game design and development course that would cover the theory behind game design, the use of a 3D game development platform, and also give students enough time to develop two complete games. However, the very steep learning curve of most 3D game development platforms prevented us from doing everything we wanted to do in one semester. Then we decided to give Alice [5] an opportunity.

In the spring of 2009, we taught a game design and development course using Alice as our 3D game development platform. The course was fourteen weeks long and was taught in the Computer Science department of New Mexico State University. This was the first time that a game design and development course was taught in our department. The course prerequisite was a data structures and algorithms course. The course was extremely popular and filled up completely on the first day of registration. We had 27 students enrolled in the course including 5 female students. In this paper, we describe our very positive experience in teaching this course with Alice as a 3D game development platform.

We were very satisfied with the quality and the variety of games produced by our students. Our students enjoyed the course very much, learned a lot, and felt a strong sense of accomplishment by having developed two complete 3D games even though none of them had developed any game before.

Alice has been used to teach how to make very simple games to middle school students [11, 22], CS0 students [13], and CS1 students [8, 18], but to our knowledge nobody has used Alice as a 3D game development platform to teach an actual game design and development course. We hope that this paper encourages other faculty to give Alice an opportunity as a 3D game development platform.

There are many 3D game development platforms available for free besides Alice, such as jMonkey Engine [12], Panda3D [15], OGRE [14], Blender [7], 3D Rad [3]. Other 3D game development platforms are available for a fee such as Torque 3D [21], 3D Game Studio [2], Game Maker with its 3D advanced features unlocked [10], Silent Walk FPS Creator [19], and 3D Gamemaker [1]. The jMonkey Engine was used in a game development course at Southern Illinois University [20]. The Torque 3D engine was used at the University of North Carolina, Wilmington to teach a game development course [16]. In these two courses, most of the semester was spent on learning the game platforms, and not enough time was left to cover a sufficient amount of game design theory. Game Maker and Blender were used on a gaming course at the University of West Georgia [17]. This course is the second of a two course sequence where the first course concentrates on game design. The gaming course was only able to produce one 3D game. Silent Walk FPS creator, 3D Gamemaker, and 3D Rad are specialized platforms...
that allow only for the creation of first person shooter games, shooting games, and vehicle simulations games, respectively.

Compared to the above platforms, Alice has an extremely easy learning curve which enables students to learn Alice very fast. This gives enough time during the semester to cover the material of a book on game design theory and still have time to develop two complete 3D games. Also, Alice allows for the creation of many different types of 3D games.

In the remainder of the paper we describe why we chose Alice as our 3D game development platform, what material was covered in our course, how we conducted the course, how we performed the grading, the quality of the students’ game projects, features of Alice that students liked for game development, problems that students had when using Alice, and the evaluation of the course.

2. REASONS FOR CHOOSING ALICE

We chose Alice as our 3D game development platform for the following five reasons.

(1) Our very positive previous experience of using Alice in another course. We have been using Alice to teach a computer animation course (with no prerequisites) at our computer science department for many semesters. Also, this course is taught to high school students during the summer as part of our Young Women in Computing summer workshop. In this course, we teach students how to program using Alice. As their final project, our students develop games, and every time that we teach this course, our students impressed us with their games. Therefore, we knew that with Alice it is possible to create interesting games.

(2) We wanted to try an experiment with our upper level computer science students who have already completed a data structures and algorithms course. Given our success with the games developed by our high school and lower level computer science students, we wanted to see the different types and quality of games that could be produced by upper level computer science students, who already know how to program, and have had some semesters of programming experience.

(3) We wanted a 3D game development platform that could be learned in a short amount of time (a few weeks) so our students could have time to learn the fundamentals of game design of a complete book, and still have time to develop two complete 3D games during our fourteen week course.

(4) We wanted a 3D game development platform where lots of different types of games could be implemented.

(5) We wanted a 3D game development platform with lots of 3D models already built-in. This was very important since we did not want our students spending a long time learning a complicated 3D modeling tool and then spending even more time drawing their 3D models.

3. MATERIAL COVERED

Two textbooks were used to teach our course: Learning to Program with Alice [9] and Game Design and Development: Fundamentals of Game Design [4]. The first textbook, was used to introduce Alice to our students. The second textbook, was used to teach game design theory.

In our first two lectures, we dedicated ourselves entirely to Alice. Then, we covered the remaining material of the Alice book for the first 15 minutes of each lecture for eight more lectures. We were able to cover the entire Alice book’s material very fast because our students already knew how to program. Even though the Alice’s book material was covered completely in the first ten lectures, we dedicated the first 15 minutes of each lecture for the remaining of the course to specific students’ questions on Alice and on their Alice game projects.

In our third lecture, we started covering the material of the game design book on the third lecture. We did not lecture. Our students were the ones who gave presentations on each chapter from the book.
3. GRADING
Several criteria and different grade percentages were used to grade the students in the course: oral presentation (30%), game project1 (10%), game project2 (30%), exam1 (10%), exam2 (10%), class attendance (5%), and class participation (5%).

The exams were individual closed-book and in-class exams. The exams covered basic concept questions from the oral presentations. The main purpose for having the exams was to make sure that students paid attention to the oral presentations.

Course attendance and participation encouraged students not only to come to class, but also to pay attention to the oral presentations, and therefore be able to formulate questions, and participate during the 15 minutes of questions and discussion that followed each presentation.

For the game projects, the students were free to select their own game types, themes, storylines, characters, etc. The first project was done individually and the second one was done by a team of up to three students. The students made their own teams for the second project. The main criteria used in the evaluation of the projects were quality and effort.

All the games were presented and played in class by their developers.

We had very good final grades: we had 17 As and 10 Bs.

5. GAME PROJECTS’ QUALITY
We were a little bit disappointed with the quality of the first game project submitted by our students because the games were very similar in type and quality to the final projects developed in our lower level computer animation course. All games were either treasure hunting games (40%), avoiding obstacles games (28%), or shooting games (32%). However, we and our students were very satisfied that they were able to design and develop a complete 3D game in one month while learning Alice at the same time.

On the other hand, the second game projects were excellent. The students developed games of great quality and we were able to see games types that we had not seen before in our computer animation course. There were action–adventure games (40%), puzzle games (20%), timing and rhythm games (10%), as well as treasure hunting (20%) and shooting games (10%).

We believe that the main reason for the improvement in the quality of the second game project was that our students were able to incorporate more game design material from the game design book. At the time our students started the design of the second game projects, we had covered almost all the material of the game design book. Some of the game design features that students incorporated and improved on in their second game project include: improved user experience, clean and intuitive interfaces, games relied more on skill and strategy, challenging game play, explored different types of games, interesting stories, improved character development, and more story and play related 3D worlds.

Other reasons for the improvement in the quality of the second game project are the following. (1) The students’ creativity improved significantly when they worked on a team because they combined their ideas. (2) Since the students already had the experience of presenting their first game in front of the class, they wanted to present a better game the second time around. (3) Since we were going to give a small award to the best three games (voted by the students), the students added an extra effort. Most students in the class were male and like to compete. (4) The students already had learned and experienced Alice. (5) The second game project was worth 30% of their grade.

Figures 1 through 4 show some of the games developed by our students.

The Morph game (Figure 1) is a shape matching timing and rhythm game where the player needs to change and rotate figures to match specific shapes which appear synchronized with background music. There are three levels of difficulty in the game.
The Survival game (Figure 2) is an action–adventure game where the player is the brown dinosaur which needs to remain alive and escape from a jungle full of dangers. This is a very challenging game to play even for the students who developed it.

The Fighting game (Figure 3) is an action game where the player is the samurai who is fighting a dragon. The samurai can throw arrows to the dragon which tries to avoid them by flying away and throwing fire in return.

The website http://www.cs.nmsu.edu/~kvillave/alice/cs474 contains several of the Alice game projects made by the students. The games can be downloaded and played in Alice 2.0. All the games are memory intensive and require a computer with at least 2 GB of RAM.

6. FEATURES OF ALICE THAT STUDENTS LIKED

Among the several features of Alice that students liked for the development of 3D games, we have the following.

(1) Alice comes with hundreds of already built-in low polygon 3D models. And all the parts of the 3D models can be individually manipulated before and during run-time. Therefore, there is no need
for multiple versions of a 3D model to create an animation. Our students used only the Alice models for their game projects. They did not spend time learning a modeling tool and then creating and exporting the models to Alice.

(2) Alice provides automatic handling of computer graphics. When objects move in the virtual world, their perspective, lightning, and fog effects are modified automatically. Students can also easily adjust the ambient, directional, point, and spot lights, and fog effects during run time. Automatic handling of computer graphics allowed students to concentrate on their game design and mechanics instead of studying and coding complicated computer graphics manipulations and dealing with graphics hardware nuances.

(3) Alice provides automatic camera handling. For first-person perspective all that is required is to set the vehicle property of the camera to the character.

(4) Alice comes with a very friendly user interface especially in terms of 3D object manipulations. Re-arranging, resizing, rotating, copying, turning, and tumbling 3D objects in the virtual world is effortlessly achieved by using Alice’s quad view of the virtual world and mouse controls which also allow zooming and scrolling.

(5) Alice comes with a capturing pose feature which is very useful for object animation. This feature allows students to make an object look a certain way by manually moving its parts and capture the new looks (poses) by just clicking a button. Alice can then cycle through these poses at run-time and fill in the movement details between the poses.

(6) Alice provides a very easy to use event editor and also automatic event handling.

(7) Alice allows students to experience some basic concepts of parallel programming with its built-in DO TOGETHER instruction where blocks of code can be set to run simultaneously.

(8) Alice’s programs can be speeded up and slowed down at run-time using a scroll bar. This feature was of great help for debugging and testing our students’ game projects.

(9) Alice comes with a drag-and-drop code building feature. According to our students, this feature allowed them to pick up the language really fast.

7. STUDENTS PROBLEMS WITH ALICE

Even though our students enjoyed working with Alice very much, they did experience several problems which we list below.

(1) In Alice it is not possible to copy/paste code directly between Alice files. There is no merge feature between Alice files either. Our students had to rely only on the export/import of classes feature to share their work when they worked on their team project.

(2) Alice does not allow the creation or destruction of objects during run-time. Therefore, our students had to improvise by placing all their objects at the beginning of their game and make them visible and invisible during the game execution as it was needed. This required long load times and huge amounts of memory.

(3) Alice does not have very good sound support features. It is impossible to stop one audio file and start another one. Students had to use external audio editors like Audacity [6] to edit their sound files to just the right playing times.

(4) In Alice it is not possible to enable or disable events during run-time. Our students had to devise clever ways to work around this problem.

(5) Alice games run differently on different hardware platforms and Alice versions.

(6) Alice does not come with automatic collision detection, it has to be programmed. Also, it is not possible in most cases to program accurate collision detection since Alice only allows us to compute the distance between the centers of two objects.

(7) It is not possible to create user defined classes different than normal Alice classes.

(8) It is not possible to create two-dimensional and multidimensional arrays.
8. STUDENTS COURSE EVALUATION

The students were very happy with the course, especially because they were able to develop two complete 3D games even though they had never developed any game before. Our students enjoyed very much the students’ chapter presentations. They said that by having a different speaker in every lecture, the lectures were kept fresh, dynamic, and interesting.

Our students enjoyed seeing all the game projects in class. They enjoyed watching what other students had done, and also liked the small game competition we had for the second project where we gave prizes to the best three games voted by the students. Our students also liked participating in the questions and discussion section at the end of each oral presentation. They said that they learned to formulate questions and discuss their ideas in front of everybody.

10. REFERENCES