Abstract - Computer games are an often cited motivational tool used as assignments in introductory programming courses. This paper presents the results of a study that attempts to quantify the effectiveness of computer games as programming projects over non-games. Students completed five major programming assignments during the semester in an introductory programming course. For each assignment, students were given an option between two programming projects, a game and a non-game. Throughout the term, 78.9% of the assignments submitted by students were the game option. The average grade, however, on game assignments was 89.1%, while the non-game average was 95.1%. In a post-term survey 84% of student respondents answered “yes” to the question, “Did the games provide any extra motivation for you to complete the projects with a high level of quality?” Results suggest that games do indeed provide psychological motivation and increase course enjoyment, even though they may not improve students’ scores.

Index Terms – Computer Games, Computer Science Education, Evaluation, Pedagogy, Programming

INTRODUCTION

Most people enjoy playing games of some sort, and a large number of college students play computer games on a regular basis. Many computer science instructors have attempted to take advantage of this fascination with games by using them as examples and assignments in undergraduate computer science courses. For instance, Shifroni and Ginat [1] developed a game to help teach network communication protocols by requiring students to act out the roles of various layers of the OSI model. Baker et al. [2] created an entertaining card game that helps students learn about the software engineering process. Ariyapperuma and Minhas [3] have used freely available online games as laboratory activities to teach network security. Natvig and Line [4] developed a web-based game environment that introduces concepts in a computer fundamentals course. Other instructors have used games as an ongoing theme when teaching object oriented principles [5], and some have even incorporated special software packages into their courses that allow students to easily create sophisticated graphical games while learning about object orientation [6]-[7]. Whole courses [8]-[9] and even concentrations within the computer science curriculum [10] have focused on the idea of game design.

Games can aid in the understanding of a difficult concept or idea, and are often said to increase student motivation in programming courses [11]-[12]. Games can help make project descriptions clearer, since students often have prior experience with the games they are to program [13]-[14]. However, use of games in computer science courses can also have some negative effects. Most notably, games tend to be more appealing to the male students, and competitive games may even “turn off” some females [15].

In light of the debate over games in the curriculum, it is surprising that very few formal evaluations have been conducted to study the successfulness of approaches that teach computer science concepts through the use of games. Some notable evaluation efforts include that of Shifroni and Ginat [1] who evaluated the usefulness of their network game by comparing the exam results of students who used the game to learn network concepts, to those who did not. Feldgen and Clua [16] observed that retention rates were higher in introductory programming courses after a games approach was introduced. Baker et al. [2] evaluated their card game by having students who had taken a software engineering course play the game, and then answer questions on a survey that asked for opinions on the game’s effectiveness as a teaching tool. Natvig and Line [4] also used surveys to evaluate their web based game environment. Ariyapperuma and Minhas [3] used informal interviews to assess the effectiveness of online network security games as laboratory activities in their course. They also compared solutions for a case study between two student groups, one who had played the network games, and the other who had not.

This paper discusses the results of a study that attempts to provide a formal assessment of the effectiveness of computer games as assignments in an introductory programming course. The purpose of the study was to pursue answers to the following questions: one, do students really prefer game assignments to non-games? Two, do games assignments improve student scores? Three, is the motivation of students to learn programming skills increased when they have game assignments compared to non-games? The remainder of this paper is arranged as follows: Section two describes the structure of an introductory programming course which was used as the test environment for this study. Sections three and four present the results of the study and the responses to a survey that was given to the students at the conclusion of the course. Section five discusses the conclusions drawn from this experience.
A study was conducted during the Fall term of 2005 to determine the effectiveness of games as programming assignments in introductory programming courses. Two sections of CS 121 – Programming I were involved in the study. Both sections were taught by the same instructor and initially contained a total of 33 students (17 in section A and 16 in section B). There were nine female students and 24 males. Only one student withdrew from the course during the semester.

Students were required to complete five major programming assignments throughout the course of the term. For each assignment, the students were given a choice between two project descriptions (one a game, and the other a non-game). Care was taken to ensure that the program options for each assignment were of comparable difficulty and that they required students to implement similar programming constructs. Students were allowed to choose either program with no penalty to their grade. Their only instruction was to choose the program that seemed “most enjoyable.” Students were not told of the study during the term.

The following is a brief description of the options for each programming project. All projects were console based applications written in C++. The projects were described in a great deal more detail in the actual assignment descriptions provided to the students.

Project 1A: Number Guessing Game. The program picks a random number, X, and then displays the result of the following computations: X%5=?, and X/5=? The user attempts to guess the value for X. Three questions are given with different values for X (and 5), and the program counts and displays the number of correct answers at the end.

Project 1B: Change Making Program. The program randomly determines the cost of an item and asks the user for an amount of currency with which they wish to pay. The program then shows the correct number and type of bills that should be given as change.

Project 2A: The Marble Game. The program implements a game played with a pile of marbles (but simulated in the program with integers). The player and computer take turns selecting some number of marbles from the pile. Any number of marbles between one and half the total amount is allowed to be taken on each turn. The object is to force your opponent to take the last marble.

Project 2B: Factors of Numbers. This program asks the user for a number, and then lists the divisors for all numbers less than that value, down to, but not including 1. If a number is prime, the program states this fact.

Project 3A: The Casino Game. This program implements a simple casino. The games available at the casino are High Low, Roulette, Jackpot, and Twenty One. The program keeps track of the player’s money for each game and makes them leave the casino when they go bankrupt (or when the user gets tired of playing).

Project 3B: The Movie Theatre. The purpose of this program is to help sell tickets at a movie theatre. The theatre contains a number screens and has multiple showings for each screen. The program tracks how many tickets are sold for each showing during a day, and makes sure that showings are not oversold. The program concludes by providing a listing of ticket sales for the day.

Project 4A: Tic-Tac-Toe. The program implements a text-based version of the game tic-tac-toe between the computer and user. For full credit, the computer has to be able to play with some intelligence.

Project 4B: Blob Finder. The program allows users to enter values into a multi-dimensioned array. Once the user has finished entering numbers, the program then determines how many “blobs” are in the array. A blob is defined as a sequence of connected cells in the array.

Project 5A: Hang Man. This program implements the popular word game hang man. The program selects a word from a data file of words, and then the user attempts to guess the word by selecting letters they think might be in the word. If the user does not guess all the letters before six incorrect guesses, they lose.

Project 5B: A Bookcase. This program allows users to maintain information about a bookcase. The program reads book information from a data file and stores this data into a two dimensional array of objects (to simulate the bookcase). The program supports options to locate books and to add and remove books from the bookcase.

At the end of the term, after all projects had been submitted, students were asked to complete a questionnaire about the course assignments. The next section of this paper discusses the choices students made about which projects to attempt and their performance on those assignments. The following section discusses student responses on the survey.

RESULTS

Throughout the course of the term 120 of the 152 (78.9%) assignments submitted by students were the game option. This would appear to support a general conclusion that students prefer game assignments to non-games. There did appear to be a difference between male and female preferences. The females submitted 29 game assignments (65.9% of submissions) and 15 non-games (34.1% of submissions). The males submitted 91 game assignments (84.3% of submissions) and 17 non-games (15.7% of submissions). However, both genders did attempt more game assignments than non-games, supporting a conclusion that both genders prefer game assignments to non-games.

As the assignments became longer and more complex, the games became even more popular as 88.6% (78/88) of submissions for projects three, four, and five were the game
option. For the final three projects, females submitted 22 games (84.6% of submissions) and 4 non-games (15.4% of submissions). Males submitted 56 games (90.3% of submissions) and 6 non games (9.7% of submissions). The increased preference for games for the last three assignments could be explained by the fact that students are already domain experts [11]-[16] with the games. The game assignments for projects three, four, and five were games that practically all the students had played before or had at least seen (simple casino games, tic-tac-toe, and hangman). The non-game options likely seemed considerably more esoteric to the students, and would have required a great deal more time just to understand the project description before students could write any code. Conversely, most students felt that they could get started immediately programming the games since they already understood the associated algorithms (or at least thought they did).

Interestingly, the game assignments did not seem to improve student scores overall. The average grade on game assignments throughout the term was 89.1% (standard deviation = 15.9), while the non-game average was 95.1% (standard deviation = 11.0). There are a number of reasons why this could be the case. The author feels that, most likely, the best students in the course were the ones choosing the non-game assignments. In an informal conversation with a student early in the term, the author asked the student what criteria he was using for selecting projects. The student replied that he looked at both assignments and chose the one that he thought looked more difficult. That way, if he could figure out how to do the more difficult project, he was probably learning more in the course. This was one of the top students in the class. Given that students likely had more familiarity with the topics of the game assignments, there may have been a perception by the students that the games were “easier”.

SURVEY RESPONSES

At the conclusion of the term, after all assignments had been submitted, the students were asked to fill out a survey about the course projects. There were 25 students present in the classes the day the survey was administered. The first question asked the students, “For each project, did you like having a choice between two programs?” All 25 students responded “yes” they did like having a choice. The next question asked students to explain how they selected the project they were going to attempt. Most students answered that they picked the one that looked easier or the one that looked like it was more fun. A few responded that they chose the project they knew the most about, or the one that looked most interesting. One student wrote, “Honestly, the program I was going to do was the one talked about the most in class. So, although there was a choice, I always did the first choice.” Another student stated, “I usually went back through my notes and whatever project looked like I had more in my notes on then I chose that one.” The instructor made an effort not to bias the students towards either of the two projects in lectures, but did offer to answer questions about the projects during class time. In the instructor’s opinion, students did ask about the game assignments more often than non-games, probably because more students were working on the games. The student comments may suggest that at least some of the students chose the game assignments largely because that was what a majority of their classmates were working on as well.

Question three on the survey asked students if they felt both program options were equally well explained for each project. On this question 22/25 (88%) said “yes”, the projects were equally well explained. The students who answered “no” wrote the following:

- “It may simply have been a lack of familiarity with the subject on my part, but there was usually one program that made more sense.”
- “Sometimes I thought that one may have more code to write. But other than that it was fine.”
- “like on project 5 I did not understand what was wanted for the project. The instructions on things to include were a little unclear.”

It does not appear from the survey responses that the project descriptions had a substantial impact on how students selected which project to submit.

Question four asked students which project was their favorite. The casino project (Project 3A) was the top selection with 11 votes. Hangman (Project 5A) was second with five votes and tic-tac-toe (Project 4A) was third with four. The only non-game project to receive any direct mention was the movie theatre program (Project 3B) with two votes. Three students did not identify a specific project as their favorite. Overall, 20/25 students (80%) mentioned a game assignment as their favorite.

On question five, students were asked if they noticed that one of the two options was always a game. Only 18/25 (72%) made this observation, which surprised the author somewhat. This may suggest that the other seven students were choosing projects based on interest, and not just selecting the games. The sixth question asked students, “Did you enjoy working on game assignments?” for which 22/25 (88%) responded “yes”.

The survey did not ask students to indicate their gender, so it is unknown whether the three who answered “no” to this question are male or female.

Probably the most interesting result from the survey came in response to question 7, as 21/25 students (84%) answered “yes” to the question, “Did the games provide any extra motivation for you to complete the project with a high level of quality?” This may seem to be in contrast to the actual project scores, since the non-game assignments had a higher average grade (95.1%) than the games (89.1%). Even though the survey did not ask students to comment, one of the four who answered “no” to this question wrote in the explanation, “the fact they were projects for a grade motivated me, being a games was just extra fun.” The author believes that, as mentioned earlier, some of the best students were likely choosing the non-game options because they perceived them to be greater challenges. Therefore, non-game assignment
submissions probably represented the work of the best students in the course.

CONCLUSIONS

The study described by this paper attempted to answer the following questions: one, did students prefer game assignments to non-games; two, did games assignments improve student scores; and three, was the motivation of students to learn programming skills increased when they had game assignments compared to non-games? The answer to the first question seems to be fairly clear. Students chose substantially more game assignments than non-games over the course of the term (78.9% to 21.1%). A large majority of students also stated that they enjoyed working on game assignments (88%) in the survey, and 80% of the students mentioned a game as their favorite assignment. Thus it appears that a majority of students do prefer games to non-games. The preference for games was stronger for males than females, but both genders did select more game assignments than non-games.

The answer to the second question, did game assignments improve student scores, would appear to be “no”. Non-game assignments had a higher overall average (95.1%) than non-games (89.1%). However, 84% of the students did answer affirmatively when asked if the games did provide any extra motivation to complete the project with a high level of quality. It could be the case that the games did improve the scores for a large number of students who chose the game options. As argued earlier, the better students were likely the ones submitting the majority of non-games, and so the 95.1% average may be artificially inflated. The author feels that it could be fair to conclude that the game assignments did improve scores for a large number of students who preferred game assignments. However, this is certainly a question that deserves a great deal more study.

The answer to question three (was the motivation of students to learn programming skills increased when they had game assignments compared to non-games) also appears to be affirmative for most students. Again, 84% of the students surveyed answered “yes” when asked if the games provided any extra motivation to complete the projects with a high level of quality. Only two students answered “no” to both questions six and seven on the survey, meaning that they did not enjoy programming the games and did not find them motivating. This was a very small percentage of survey respondents (8%). The vast majority of students did seem to draw some enjoyment and/or motivation from the games, which in turn likely made the course more interesting for the students.

Probably the greatest lesson learned by the author while conducting this study was that all of the students enjoyed having a choice on the programming projects (survey question #1). The author now plans to provide students with at least two options for each major assignment in future introductory programming courses. The preparation required to teach a course with this option is slightly more (two projects must be prepared instead of one for each assignment), but the grading load is no more. Providing students with a choice between a game and a non-game seems to offer the best of both worlds. The students who enjoy games and find them motivating are provided with this option. Any students who do not want to work on games do not have to. This seems a fair way to deal with concerns that game assignments discriminate against females in introductory programming courses [15]. Offering students a choice empowers them, and this provides motivation for both genders.

REFERENCES

