Game2Learn: Improving the motivation of CS1 students

Tiffany Barnes, Eve Powell, Amanda Chaffin
University of North Carolina at Charlotte
Computer Science Department
9201 University City Blvd, Charlotte, NC.
tbarnes2@uncc.edu

Heather Lipford
University of North Carolina at Charlotte
Software and Information Systems Department
9201 University City Blvd., Charlotte, NC
Heather.Lipford@uncc.edu

ABSTRACT
Games are increasingly being used for education and training in a variety of areas. We are developing a game to teach introductory computer science concepts, called Game2Learn, to increase student motivation and engagement in learning CS1, which are critical for recruiting students into computer science. We evaluated student feedback and performance of initial prototypes to examine the Game2Learn concept and provide design guidelines for ongoing game development. In this paper, we present the results of this study, which demonstrate that students can have fun programming within a game, and that in-game rewards and punishments are vital to the motivation and potential learning of students.

Categories and Subject Descriptors
K.3.2 [**Computers and education**]: Computer and information science education. – computer science education.

General Terms
Design, Human Factors

Keywords
Game development, education, motivation, evaluation.

1. INTRODUCTION
Although demand for computing jobs is rising, the number of students enrolling and graduating in computer science is declining, while the proportion of women and minorities is also decreasing [1]. Attrition rates in computing are as high as 30-40%, with most students leaving after taking either CS1 or CS2 [2]. Reasons for these high attrition rates include: poor knowledge about computing, poor math and problem-solving skills, poorly designed lab courses, lack of practice and feedback, and under-prepared instructors [2]. On the other hand, games are increasingly being recognized to have built-in motivation and familiarity for most students [3-8], as well as incorporating expertise from a variety of computer science-related fields [9]. Increasingly, researchers are looking to leverage games to engage and motivate students in computer science, and most of these approaches incorporate building games [10-13].

GDCSE '08, February 28-March 3, 2008, Miami, FL, USA.

Game2Learn, on the other hand, seeks to improve recruiting and retention in computer science through immersing computing instruction in game-based learning environments. We hypothesize that teaching introductory programming through playing a game can improve student engagement, motivation, and learning. We focus primarily on engagement and motivation, because these factors, according to researchers, may be the most important factors in learning [4].

The development of a fun game that is also instructionally effective, and that creates a motivated learner is a difficult and complex process [4]. We have complicated this equation by engaging senior undergraduate and research students in creating Game2Learn games as reported in [14]. Therefore, we are employing a spiral software development cycle, alternating rapid prototypes with evaluation to investigate the effectiveness of our developed instructional games. In this paper, we present two prototype games that demonstrate the Game2Learn concept, along with evaluations of the effects of these games on student motivation and engagement.

After our first cycle of development and evaluation, we observed that students made little connection between in-game performance and learning computer science concepts, and that they had low motivation to complete the games with few learning errors. Student comments led us to revise our prototypes to provide more explicit rewards and punishments for right and wrong answers and then continue our evaluation on these new prototypes. We found a marked difference in perceptions and attitudes towards the games, as well as some improved learning effects after incorporating more explicit goals and immediate feedback. These results demonstrate the importance of feedback in motivating the user to learn in a game, and the effects on performance such rewards and punishments may have.

In the rest of the paper, we first briefly discuss related work on educational games and feedback. We then introduce our prototypes and evaluations. Finally, we present theresults of these evaluations and their implications on providing in-game feedback in Game2Learn and other educational games.

1.1 Background
Games are becoming increasingly recognized for their inherent motivation, as inspiration for improving educational applications [4,5]. However, there is little consensus on what makes for an effective instructional game [4]. Lepper and Malone have investigated the most important factors in making educational games fun, listing as most game design books do the importance of challenge, or the balance between ease and difficult, in engaging learners in games, but highlight the need to design in activities that help learners address and revise their misconceptions [7]. Garris, Ahlers, and Driskell [4] have examined the literature on instructional games and classified the

factors that are important to their effectiveness for learning, and have identified motivation to play and play again as a key feature of the best instructional games. Their framework of considering cycles of "user judgments, behavior, and feedback" [4] reveals that is important for game engagement to be woven in with the types of feedback most useful in learning. We have therefore striven to design an educational game that balances play and learning time, makes strong ties between in-game motivation and learning outcomes, and that can be used as learning tool in a standard introductory computing course.

2. GAME2LEARN PROTOTYPES

To balance the need for formative feedback to ensure success, and to accommodate small development teams, we have chosen to employ a highly iterative rapid prototyping development model [14], and to use existing game engine technologies to provide content including art, models, and sounds. To begin the process of game development, we chose concepts from the IEEE and ACM joint curriculum for CS1 [16], including conditionals (if-then), iteration (for and while loops), and recursion. We created two very different prototype games around these concepts to explore various game possibilities.

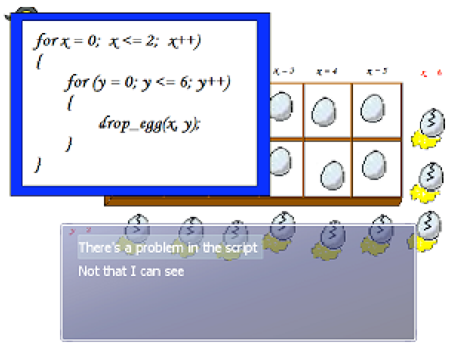

Figure 1. The egg drop quest in Saving Sera.

"Saving Sera" is a two-dimensional exploratory game, implemented using RPGMaker [17], where the player learns of the kidnapping of the princess and determines to rescue her. However, Sera is far from helpless, and in fact plays a key role in the final battle of the game. The user must perform various tasks involving programming concepts: correctly reordering a while loop statement of a confused old fisherman's mind; correcting a nested for loop placing eggs in crates; and visually piecing together a quicksort algorithm. When the player makes a mistake, the character must fight a script bug, which asks the users various computer science questions in order to fight the bug.

"The Catacombs" is a three dimensional game developed using the BioWare Aurora toolset [18] that was used to build NeverWinter Nights, a popular fantasy role-playing game based on Dungeons and Dragons. In this game, the user is an apprentice wizard who must perform three progressively more complicated tasks to save children stuck in the catacombs. The first involves unlocking the door to the catacombs involving two if statements; the second, building a bridge brick by brick with nested for loops; and the third, solving a cryptogram using more nested for loops. In the second and third quests, incorrect answers resulted in decreasing player health. The game uses dialogue with a sarcastic spellbook named Grimore and multiple choice questions (a dialogue tree) to create the code and complete the tasks.

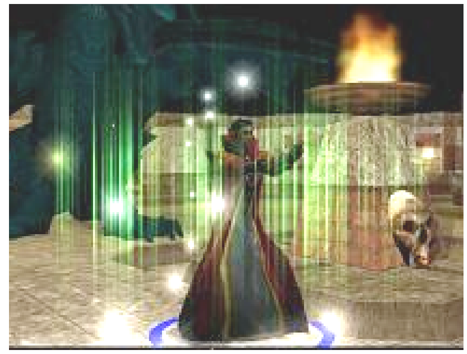

Figure 2. Casting a spell in The Catacombs.

3. METHOD

We performed two exploratory evaluations of our prototypes with students who had already taken a CS1 course. Study 1 was a pilot study to investigate the feasibility of using such games for homework, and to gather formative feedback on the games. Study 2 was a followup study to evaluate changes made based on Study 1 results. Study 1 was conducted in August 2006 and Study 2 was conducted in early September 2006.

In each study, participants signed an informed consent form, took a demographic survey and a pre-test of computing concepts, played about 20 minutes of each game, including Saving Sera and The Catacombs (where each game contained three quests), took a post-test, and were interviewed about their experience. The demographic survey includes information on the participant's race, gender, year in school, major, and gaming habits. The post-test is similar to the pre-test, but with the questions rearranged, and the numbers and variable names changed in each question. Each session took about one hour. In Studies 1 and 2, participants were interviewed about each game immediately after playing, and the post-test was given afterwards; gameplay and interviews were videotaped.

Semi-structured interviews were used to gather what testers thought of using games like ours as homework, which games and/or quests they preferred, and how balanced the games were in play to quest time. Our primary outcome measurements were the qualitative responses about whether 1) the games were enjoyable, 2) subjects felt they could learn with them, and 3) subjects would prefer to learn using a game such as these. In Study 2, we also asked subjects to report on how the game structures of rank and experience points for quests affected their motivation. Gameplay videos were analyzed for problems in using the game interfaces, and for patterns and differences in user behavior. The interviews were transcribed and coded to determine areas for game improvement. Demographic and pre-test/post-test results were used to categorize responses to see if different groups responded differently to the games.

4. RESULTS & DISCUSSION

4.1 Study 1 Results

Our initial evaluation included 13 students from a variety of backgrounds and computer experience, including 2 Asian females, 2 white females, 1 black male, 1 Hispanic male, and 7 white males. Most were ages 18-24, with 2 participants ages 25-30 and 1 aged 31-40. Nine participants were in computing related majors while the other four were studying electro-mechanical systems, psychology, communications, and art. All participants but one had taken a CS1 course, with most of these taught in either Java or C/C++. Participants included 5 sophomores, 4 juniors, 2 seniors, and 2 college grads.

Overall, we received positive comments from participants about our prototypes, providing initial evidence that we can achieve our goals of providing an engaging learning environment. The 10-question pre- and post-tests asked students to predict outcomes for short conditional statements, for loops, and a recursive function, and to set stopping conditions in for loops. Although we did not expect significant learning gains over the study, scores were still surprisingly low, with the nine computing majors averaging pre: 6.44 / post: 6.56 and non-majors pre: 3 / post: 2.67. Despite poor test results, students completed most quests, and were overwhelmingly enthusiastic about the idea of learning in games. Most were also enthusiastic in suggesting changes for future Game2Learn games.

Of the 13 participating in the study, we have 7 full interviews recorded and transcribed. Five of these reported being "core" gamers who play often, while two were casual gamers who rarely play games. In the interviews, a majority of participants found both games fun, but a few seemed to find Saving Sera more fun and less confusing. Participants appreciated being about to learn while playing in a familiar game setting and ranked both games as slightly easy. Several students enjoyed creating code in each game, but a few felt they needed more teaching before playing.

We did not anticipate one effect of Study 1: some students approached the games just as games, while others viewed the games as learning tools, and both sides questioned the seriousness of the games for homework. Some students seemed to think of our prototypes as mainly a game, and enjoyed playing such a game involving programming, but some were unsure if the game tasks would teach them enough as part of a course. Other students thought of the prototypes as potential homework, and thought a game would make homework more fun but questioned whether the game tasks would be serious enough assignments, especially since students could guess and eventually get correct answers on the game tasks. Additionally, we noticed that a number of students sometimes did not read task instructions or other game screens very carefully, leading to errors in performance. Thus, not only did students question the potential seriousness of the game, they also did not always take the game seriously themselves, despite performing poorly.

4.2 Study 2 Results

We realized that in our initial prototypes, we did not provide very clear feedback that a player's health or battles were tied to correctly performing learning tasks. If this feedback is appropriate, whether students approach the tasks as a game or as homework, the feedback should be relevant and motivating. This led us to modify our prototypes to add more explicit rewards and punishments and tie feedback more closely to the learning goals. We then continued our evaluation with these new prototype versions to see if there were any changes in the behavior or comments of students.

To Saving Sera we added a ranking system, where players start at rank 7, and aspire to become rank 1. The rank is always visible on the screen. Players with higher rank receive bonuses in the game, such as extra information from non-player characters and discounts in the shops. To The Catacombs, we added an immediate feedback message and penalty for each incorrect answer in the game. The penalty for an incorrect answer is a 20% decrease in experience points for the quest, and after 5 mistakes, the player is temporarily turned into a chicken. We also added an initial cut scene that introduces the game, as well as a final scene that summarizes the player's performance.

We hypothesized that the added feedback would lead to more motivation to correctly perform tasks the first time, making fewer errors in each task, and that performance would increase between pre- and post-tests. We also hypothesized that users would have fewer concerns regarding the seriousness of the games to be used as homework in a course.

After incorporating these changes into the Game2Learn games during September 2006, we ran another usability study (Study 2) with 8 participants, 7 males and 1 female, using the same protocol. Five participants were computer science majors, while three were in other disciplines. Again, most were aged 18-24, while one student was 25-30 and two were 31-40. Three participants were sophomores, three juniors, one senior and one graduate student.

Our results were positive, showing that while 54% of subjects in Study 1 felt games could be used as homework and 37% felt there was a good balance between quest and play time, 88% of the participants in Study 2 agreed that the games could be used for homework and 88% of the participants liked the play-quest balance or requested more quests. None of the Study 2 participants questioned the seriousness of the games. Although in-game performance (on time or errors) was not significantly affected, students completed a few tasks faster, and their scores tended to improve more between the pre- and post- tests. We believe the in-game feedback provided better motivation, engagement, and reinforcement of learning outcomes.

4.3 Study Comparison

We collected several data points regarding student performance with the games: pre/post test scores, incorrect responses in the game, and time spent on various portions of the game. Table 1 summarizes the pre/post test scores for each set of participants. Study 1 participants were our original subjects, while Study 2 participants had the additional in-game feedback. For all groups, there were differences in pre- and post-test scores in Study 2, while Study 1 showed no learning gains over the experiment. Although our sample sizes are too small to generalize, these findings do indicate a possible increase in learning due to the added motivational elements in Study 2. There seem to be similar learning effects for both majors and non-majors.

	Study 1			**Study 2**		
	Pre	*Post*	*Diff.*	*Pre*	*Post*	*Diff.*
Majors	6.44	6.56	0.11	7	7.4	0.4
Non-majors	3	2.67	-0.33	4	4.67	0.67
Avg.	5.58	5.58	0.00	5.88	6.38	0.50
St. Dev.	3.41	3.67		2.96	2.92	

Table 1. Average pre-test, post-test, and differences in scores in Studies 1 and 2, for majors, non-majors, and all students.

Tables 2 and 3 summarize the average time spent and number of incorrect answers for each quest in The Catacombs and Saving

Sera, for each set of study participants. We did not include the final quest in the Saving Sera game in this table, as this quest had multiple parts, and most participants did not fully complete the quest, either because of time or difficulty.

	Quest 1		Quest 2		Quest 3		Overall	
	Time	*#*	*Time*	*#*	*Time*	*#*	*Time*	*Q*
Study 1	2:33	2	2:39	2.6	2:10	1.2	21:08	2.8
Study 2	2:55	2.9	2:08	2.14	1:51	1.7	18:49	2.75

Table 2. Catacombs performance, showing average time to complete and average # of errors per quest, and Q, the average number of quests completed.

Despite poor pre- and post- test results, students were able to understand and complete most of the various programming quests, although with some mistakes being made. Many students spent 15 to 20 minutes per prototype, and were stopped due to time constraints. In general, subjects spent more time than expected on exploring the game world, interacting with characters, and learning the game's controls. Thus, students appeared to be engaged enough in the prototypes to spend more than 40 minutes playing a game involving programming.

	Quest 1 (Egg)		Quest 2 (Fish)		Overall	
	Time	*# err*	*Time*	*# err*	*Time*	*Quests*
Study 1	3:17	1	3:19	0.9	23:49	1.55
Study 2	3:40	1	1:59	0.9	22:49	1.88

Table3. Saving Sera performance, showing average time to complete and the average # of errors per quest, and the average # of quests completed.

We see a trend in increased performance between pre and post-test scores for Study 2 participants. Specifically, in the first study, 3 participants improved their post-test score, 5 stayed the same, and 5 actually performed worse. Yet in the second study, 5 improved, 1 stayed the same, and only two performed worse. We believe that the differential performance for Study 2 participants can be directly attributed to increased feedback and in-game motivation, which could have affected post-test results either through in-game learning or increased motivation to perform well on the post-test by making a connection between the tests and the in-game learning.

Our other performance measures do not show significant changes between the two study groups. However, we see an interesting trend, though not statistically significant, where users are completing the quests more quickly. The second and third quests in the Catacombs, and second quest in Saving Sera are each showing this trend. Thus, while for the most part, students are not answering questions more correctly, they may be working more efficiently due to their increased attention. While in-game correctness was not affected except in one task, perhaps the added motivation did lead to more engagement or attention to the tasks, leading to more efficient behavior and better performance on the post-test.

4.4 Qualitative Results

The student interviews and observations provided much stronger evidence that Game2Learn could be successful in improving student engagement and motivation.

Overall, subjects enthusiastically encouraged us to continue development. Students liked both Saving Sera and The Catacombs overall, and found the games well balanced between easy and difficult tasks and between play and quest time. As noted in [4,7], challenge, or keeping the game at a level that is neither too easy nor too hard for players can be difficult to achieve in a game. Several students expressed that the games may still be challenging for very beginners. However, we feel this adds to player motivation to learn both in and out of the game environment. Although students felt that more instructions were needed for some quests, they "could see using the game as a re-enforcer for a class, something to do before a test," while another student said, "Yeah! I mean, it would be awesome if like, after a lecture, the professor just said 'Alright, get to level 43 this weekend.' [He laughs.] I would have definitely wanted to be in that class. You guys should develop this into a game that's like that."

We did not see any strong patterns of game preferences, although several students did comment that the Saving Sera prototype seemed more appealing to younger students or women. The Saving Sera game controls were easier to learn, and the graphics more simple. We did not see any particular negative feelings about the Catacombs, aside from being lost at first, from any group of participants. We received positive feedback on Grimore's sarcasm and often observed players laughing throughout the game. Thus, while there may be differences in game preferences for different types of students, or for men and women, we can not yet determine how much variety we will need to offer to have broad appeal.

Another issue that we uncovered is how students approach to the game, whether as a game or as homework, affected their comments. One student stated, "It's something other then mindless clicking. You actually have to think, something rarely seen in games today." This and a few other students seemed to think of our prototypes as mainly a game, and enjoyed playing such a game involving programming, but some were unsure if the game tasks would teach them enough. Students saw the potential for learning, though, stating, "Coding was easier, but still got harder as I went." Other students thought of the prototypes as potential homework, and thought a game would make homework more fun but questioned whether the game tasks would be serious enough assignments, especially since students could guess and eventually get correct answers on the game tasks.

These comments led to the game modifications to provide stronger rewards and punishments related to the programming tasks. This added feedback led to a small change in pre- to post-test performance and big change in student comments. In Study 1, several students commented on the seriousness of using a game as homework, and only 54% said that a game could be used to learn how to code. Yet in the second study, 88% said a game could be used for learning to program, and no one questioned the seriousness of the game. Instead, these students commented on what kind of learning was more appropriate for the game. Specifically, students expressed that the game was more appropriate for learning to modify code, but questioned whether a game could contain the complexity of tasks where one learns to create new code. Another student commented that "It's not good as a learning tool, but good as a testing tool." And indeed, the punishment of removing experience points in The Catacombs for incorrect answers is more similar to taking a test. This feedback is reasonable given that our prototypes only had multiple choice questions and simple code rearrangement and did not involve creating code from scratch (although we feel we can scale the game appropriately in the future). Thus, these participants did seem to take the punishments and rewards seriously enough to consider the best use for this tool in the classroom, as opposed to whether the game belonged there at all. We believe this is a crucial component to student acceptance of games for learning tools.

Another interesting difference in comments was in the balance of quest (task) and other play times. . In the first study, 30-38% of the students thought this was balanced and the other participants requested less quest time. Yet in the second study, 63% thought quest and play time were balanced, and 2 of the remaining 3 actually requested more quest time. This again implies that the students found the programming tasks to be both engaging and useful, and with additional in-game feedback, even fun!

5. CONCLUSION

We have presented two stages of our rapid development process to develop Game2Learn games to teach CS1 concepts. Overall, students found these games engaging, but until we provided clear goals with appropriate in-game feedback, students we not inclined to use the games for learning, or to pay much attention during or after the game to learning objectives. On the other hand, our results indicate that tying game performance to learning objectives can improve student attitudes and engagement, which are two major components of learning.

The results of our two evaluations emphasize the importance of appropriate feedback, particularly in our case in how seriously the users considered the learning objectives of the game, and in motivating students to stay engaged enough to learn. The addition of feedback affected both learning and our participants comments on whether the game would be appropriate to use to learn how to program. Due to our small sample size, further studies will need to be conducted to confirm these results. We would also like to determine what effects different types of in-game feedback, such as scores, bonus items, or secret clues, have on student motivation and engagement. We continue to develop Game2Learn games and explore their effectiveness for learning programming concepts in a game environment.

6. ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation Grants No. 0552631 and 0540523, the UNC Charlotte Diversity in Information Technology Institute, and the Computing Research Association Distributed Mentor Project in 2006.

7. REFERENCES

[1] Vesgo, J. Continued Drop in CS Bachelor's Degree Production and Enrollments as the Number of New Majors Stabilizes. *Computing Research News, Vol. 19,* No. 2., March 2007.

[2] Beaubouef, T. and Mason, J., 2005. Why the high attrition rate for computer science students: some thoughts and observations. *SIGCSE Bulletin* 37, 2 (Jun. 2005), 103-106.

[3] Becker, K. 2001. Teaching with games: The Minesweeper and Asteroids experience. *The Journal of Computing in Small Colleges Vol. 17*, No. 2, 2001, 22-32.

[4] Garris, Ahlers, and Driskell. 2002. Games, motivation, and learning: a research and practice model. *Simulation & Gaming, Vol. 33*, No. 4, 2002, 441-467.

[5] Gee, J., 2003. What video games have to teach us about learning and literacy. *ACM Computers in Entertainment. 1*(1), 20-20.

[6] Gumhold, M. and Weber, M., Motivating CS students with game programming. In *Proc. Intl. Conf. on New Educational Environments (ICNEE),* (Neuchatel, Switzerland, Sep. 2004).

[7] Lepper, M. R., & Malone, Th. W. 1987. Intrinsic motivation and instructional effectiveness in computer-based education. In R. E. Snow & M. J. Farr (Eds.), *Aptitude, learning, and instruction: Vol. 3. Conative and affective process analyses* (pp. 255-286). Hillsdale, NJ: Lawrence Erlbaum.

[8] Prensky, M. *Digital Game-Based Learning*, New York, McGraw-Hill, 2001.

[9] Jones, R. Design and implementation of computer games: a capstone course for undergraduate computer science education. *SIGCSE Bull.* 32, 1 (Mar. 2000), 260-264.

[10] Bayliss, J. & S. Strout. Games as a "flavor" of CS1. In *SIGCSE2006*. ACM Press, New York, NY, 500-504.

[11] Clua, E., B. Feijó, J. Schwartz, M. Graças, K. Perlin, R. Tori, T. Barnes. Games and Interactivity in Computer Science Education. Panel at *SIGGRAPH*, Boston, MA, August 2006.

[12] Parberry, I., M. Kazemzadeh, T. Roden. The art and science of game programming, *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, March 03-05, 2006, Houston, Texas, USA.

[13] Parberry, I., Roden, T., & Kazemzadeh, M. Experience with an industry-driven capstone course on game programming: extended abstract. *SIGCSE 2005:* p91-95.

[14] Barnes, T., H. Richter, E. Powell, A. Chaffin, A. Godwin. Game2Learn: Building CS1 learning games for retention. *ITiCSE2007*: 121-125.

[15] ACM/IEEE-CS Joint Curriculum Task Force. *Computing Curricula 2001*. Accessed Sep. 8, 2007. http://acm.org/education/curric_vols/cc2001.pdf

[16] Enterbrain Corporation. *RPGMaker XP*. Accessed Jan. 2, 2008. http://www.enterbrain.co.jp/tkool/RPG_XP/eng/

[17] Bioware Corporation. *Aurora Neverwinter Toolset*. Accessed Jan. 2, 2008. http://nwn.bioware.com/builders/