Games for CS Education: Computer-Supported Collaborative Learning and Multiplayer Games

Andrea Nickel
UNC Charlotte
9201 University City Blvd
Charlotte, NC
anickel1@uncc.edu

Tiffany Barnes
UNC Charlotte
9201 University City Blvd
Charlotte, NC
tiffany.barnes@uncc.edu

ABSTRACT

Today's Millennial students have changing preferences for education and work environments that negatively affect their enrollment and retention rates into university computer science programs. To better suit these preferences, and to improve CS educational techniques, teaching methods and tools outside of the traditional lecture sessions and textbooks must be explored and implemented. Currently, both serious games and collaborative classroom work, including pair programming, are the focus of studies meant to do just this. The proposed work deals with both serious games and student collaboration research, positing that educational games with collaborative elements (multiplayer games) will take advantage of the benefits offered by each of these areas, resulting in an educational game that demonstrates increased learning gains and student engagement above that of individual learning game experiences. Collaborative educational games and software also have the potential to solve many of the problems that collaborative work may pose to course instructors in terms of helping to regulate and evaluate student performance.

Categories and Subject Descriptors

K.3.1 [**Computers and Education**]: Computer Uses in Education—*Collaborative Learning*; K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer science education*

General Terms

Design, Human Factors

Keywords

Serious games, multiplayer games, pair programming, computer-supported collaborative learning

1. INTRODUCTION

Enrollment and retention rates of undergraduate students in university computer science programs, although gradually increasing, are still low, and remain a source of concern for educators and researchers wishing to ensure the future of scientific computing research and the industry [15]. A re-evaluation of current, traditional instructional systems, students' learning and working preferences, and the development of different teaching methods ideally results in educational techniques better suited to students' needs, leading to further increases in CS program enrollment and improved CS education.

Traditional educational methods may include lecture sessions, lab sessions, and individual programming assignments, in addition to exams and other standard means of academic assessment. However, the exclusive use of such methods may not be ideally suited to today's students, particularly those in the generation born after 1982, or "Millennial students," as termed by education researchers. Millennial students prefer hands-on learning activities, and collaboration in education and the workplace. Female, African American, Hispanic, and other underrepresented students may also be inclined toward ways of learning and working that involved more group work and social interaction than traditional computing education provides [14].

To appeal to these students, several practices of computer science education, and perceptions of IT careers must be altered. Students commonly perceive IT careers as solitary ones, a view reinforced by the practice of assigning individual projects in the first courses of CS programs, and this may not appeal to those students who prefer greater social interaction [14]. Likewise, for students expecting to learn how to develop applications similar to the technologies they use every day, the simple programming assignments in introductory classes may seem to have no connection to real practice, and therefore be considered tedious or uninteresting by students.

With these problems in mind, this proposal focuses on bridging several relevant areas of research. First is the study of serious games used as educational tools. Student interest in learning computer science curriculum material increases through games' inherent motivational qualities, perhaps more so than through textbook chapters or lectures [2]. Games can also promote creative thinking processes that develop general problem-solving skills, and result in students' increased self-motivated learning outside of the classroom. The second research area is that of collaborative work in the classroom. Increased collaboration among students may

FDG 2010, June 19-21, Monterey, CA, USA

be helpful in changing the aforementioned perception of IT careers from lonely jobs to highly collaborative ones. For example, pair programming has been shown to result in the improved performance and retention of students [11]. Finally, the growing field of computer-supported collaborative learning has contributed research toward tools for cooperative software development, and has just begun to touch upon how collaborative games may aid in education.

2. PREVIOUS WORK

The premise driving educational games development is that games can provide novel learning activities that students will find entertaining, engaging, and motivating [2], which is intuitive to some extent, although surprisingly few papers offer supportive data and conclusions taken from rigorous scientific evaluations of educational game systems [6]. Not only is there a need to ensure that the effects of serious games on learning are correctly evaluated, which is relevant to the present work if not its focus, but there is also need for further design guidelines development for successful educational games. There is little published research on multiplayer educational games. One study examines how existing massive multiplayer online role playing games (MMORPGs) may be used to facilitate the learning of a second language, as for English as a Second Language (ESL) students [12]. That study shows that social interaction and collaboration can support learning in games, though the research proposed in this abstract is concerned with multiplayer games specifically designed to include educational content, unlike most existing MMORPGs.

Collaborative work and its closer approximation of real practice can positively impact students' perceptions of IT and thereby increase their willingness to enroll in technology programs [3, 13, 14]. Published empirical evidence showing the positive effects of classroom collaborative work, like pair programming, is plentiful. Collaboration can improve enrollment and retention, student perceptions of the work and of the computing field, and student performance in CS classes. Preston's review of the literature on pair programming presents common attributes of successful collaborative learning, including the presence of cooperative behavior, and positive interdependence and individual accountability among group members, and criteria for learning tasks appropriate for student collaboration, such as sufficiently complex tasks where the mastery of material is important and problem solving or creative thought is desirable. These should be taken into account when designing a serious game's challenges and evaluation system [11].

Currently, research into the combination of serious games and collaborative work (for example, collaborative, or multiplayer educational games) is an underexplored area, although recently, computer-supported collaborative learning (CSCL) researchers have begun investigation into how games are designed to support effective and fun collaboration between players. Studies on social interaction in online games like There or Second Life, [5], or World of Warcraft [10, 1] reveal how multiplayer games allow players to use in-game objects to collaboratively create new activities around them, and how social interaction in the games is facilitated and evolved. From these studies, we learn how to create multiplayer games that effectively support, or even require, collaboration between players. We also look at existing research into collaborative learning and programming tools developed for students. GREWPtool (Groupware Research in Education and the Workforce Project) [7], RIPPLE (Remote Interactive Pair Programming and Learning Environment) [4], and Jazz [9] are various tools that support synchronous, collaborative programming among groups of students. Lessons learned from the implementation and evaluations of these tools are valuable for the development of multiplayer games meant to teach computer science concepts and collaborative programming.

3. RESEARCH GOALS

The proposed research explores how applications can be developed for the classroom to take advantage of the benefits offered by both educational games and collaborative work. We hypothesize that an educational game that incorporates collaborative elements will show learning gains and increased student engagement above those for a single player educational game. Although pair learning has been shown to be effective, instructors might be reluctant to use it because they believe it will have a higher support overhead, encourage cheating, or make it difficult to assess individual students [8]. Collaborative educational games could potentially solve these problems by removing the organizational work from the instructor, and helping to regulate and evaluate student performance.

We plan to conduct an initial study to test whether a collaborative game to teach backtracking can be more effective than a single player game. We will modify Canis Ex Machina, a game we created for a games course, where a player races against a computer opponent to collect the highest number of resources in a certain amount of time. For this initial study, the game will be modified so that players control their own game agents through use of the backtracking algorithm, but in two-player mode, students will be required to coordinate their efforts in order to win. Communication between players will occur within an integrated, networked chat system, and players will also have the ability to view each other's game screens. To test our hypothesis, we will compare the single-player and two-player modes for learning gains, frequency and effectiveness of communication, and student perceptions of the game and of working with peers.

Once we understand the effects of cooperation and collaboration in Canis Ex Machina, we plan to conduct studies to further understand the effects of different forms of collaboration and interaction, such as cooperative play versus competitive play, on learning computing concepts in multiplayer educational games. Our goal is to develop guidelines for how best to support learning in computer science through multiplayer interaction. We may later be able to develop agents or adaptive strategies to regulate student interactions and determine their quality, intervening if necessary, to educate students in effective methods of communication and afford them the full benefits of working collaboratively. We hypothesize that such collaborative learning games will encourage students to build stronger peer relationships and networks, that may positively influence student performance, confidence, and retention. This work also promises some interesting technical challenges, particularly in discovering best practices for developing collaborative ed-

ucational games, how to make the games easily available for students at school or at home, and how to facilitate the formation of peer networks based around the collaborative applications.

4. CONCLUSIONS

Traditional methods of computer science education, and the perception of the computing field and industry they foster, are not well suited to Millennial students or underrepresented students who prefer more hands-on and social ways of working in the classroom and the workplace. Emerging innovations in computing education in serious games, pair programming, and computer-supported collaborative learning research will help reverse the trend of declining enrollment and retention rates. We believe that collaborative educational games have the potential to transform educational experiences for both students and educators in computer science.

5. REFERENCES

[1] S. Bardzell, J. Bardzell, T. Pace, and K. Reed. Blissfully productive: grouping and cooperation in World of Warcraft instance runs. In *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 357–360, New York, NY, USA, 2008. ACM.

[2] T. Barnes, E. Powell, A. Chaffin, and H. Lipford. Game2learn: improving the motivation of CS1 students. In *GDCSE '08: Proceedings of the 3rd international conference on Game development in computer science education*, pages 1–5, New York, NY, USA, 2008. ACM.

[3] S. B. Berenson, K. M. Slaten, L. Williams, and C.-W. Ho. Voices of women in a software engineering course: reflections on collaboration. *J. Educ. Resour. Comput.*, 4(1):3, 2004.

[4] K. E. Boyer, A. A. Dwight, R. T. Fondren, M. A. Vouk, and J. C. Lester. A development environment for distributed synchronous collaborative programming. In *ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education*, pages 158–162, New York, NY, USA, 2008. ACM.

[5] B. Brown and M. Bell. CSCW at play: 'There' as a collaborative virtual environment. In *CSCW'04*, pages 350–359, Chicago, Nov. 2004. ACM.

[6] R. T. Hays. The effectiveness of instructional games: A literature review and discussion. *Technical Report 2005-004, Naval Air Warfare Center Training Systems Division*, 2005.

[7] T. J. Hickey, J. Langton, and R. Alterman. Enhancing CS programming lab courses using collaborative editors. *J. Comput. Small Coll.*, 20(3):157–167, 2005.

[8] N. Jacobson and S. K. Schaefer. Pair programming in CS1: overcoming objections to its adoption. *SIGCSE Bull.*, 40(2):93–96, 2008.

[9] A. Meneely and L. Williams. On preparing students for distributed software development with a synchronous, collaborative development platform. In *SIGCSE '09: Proceedings of the 40th ACM technical symposium on Computer science education*, pages 529–533, New York, NY, USA, 2009. ACM.

[10] B. Nardi and J. Harris. Strangers and friends: collaborative play in World of Warcraft. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 149–158, New York, NY, USA, 2006. ACM.

[11] D. Preston. Pair programming as a model of collaborative learning: a review of the research. *J. Comput. Small Coll.*, 20(4):39–45, 2005.

[12] Y. A. Rankin, M. McNeal, M. W. Shute, and B. Gooch. User centered game design: evaluating massive multiplayer online role playing games for second language acquisition. In *Sandbox '08: Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, pages 43–49, New York, NY, USA, 2008. ACM.

[13] K. M. Slaten, M. Droujkova, S. B. Berenson, L. Williams, and L. Layman. Undergraduate student perceptions of pair programming and agile software methodologies: Verifying a model of social interaction. In *Proceedings of the Agile Development Conference (ADC'05)*, pages 323–330. IEEE, July 2005.

[14] L. Williams, L. Layman, K. M. Slaten, S. B. Berenson, and C. Seaman. On the impact of a collaborative pedagogy on African American millennial students in software engineering. In *29th International Conference on Software Engineering (ICSE'07)*, pages 677–687, Minneapolis, May 2007. IEEE.

[15] S. Zweben. 2007–08 Taulbee Survey. *Computing Research News*, 2009.