Engaging Students through Mobile Game Development

Stan Kurkovsky
Department of Computer Science
Central Connecticut State University
1615 Stanley St., New Britain, Connecticut 06050
1-860-832-2720
kurkovskysta@ccsu.edu

ABSTRACT
This paper describes using mobile game development as a motivational tool to engage students early in the curriculum. Mobile devices have become an integral part of everyday lives of modern students; using these devices as a part of the coursework may help them see the immediate connections between Computer Science and real-world technology. Compared to traditional game development, programming mobile games is less complex, which enables students with limited programming experience to create playable mobile games within the scope of a single course. Experience presented in this paper may be easily duplicated, but it may be especially useful in the first college-level course for students with CS AP credits.

Categories and Subject Descriptors
K.3.2 [**Computers and Education**]: Computers and Information Science Education – *Curriculum*; K.8.0 [**Personal Computing**]: General – *Games*

General Terms
Design, Experimentation, Human Factors

Keywords
Mobile game development, curriculum, motivation.

1. INTRODUCTION
A modern college student would hardly be able to imagine living a life without spending a considerable amount of time with a mobile phone: talking, browsing the web, texting, or playing games. Mobile gadgets of several years ago are converging to combine the functionality of a multimedia player, camera, and a communication and navigation device. Students increasingly associate computing technology with these devices rather than with desktop computers, which predominantly are used to teach the vast majority of Computer Science (CS) courses. We believe that there is a good opportunity to use ubiquitous mobile devices as a motivational tool used to stimulate student interest in CS.

Ongoing enrollment crisis stems, among other factors, from the decreased appeal of CS as an academic discipline or a career choice. Furthermore, many CS students end up changing their majors after CS1 or CS2 because they find the course material irrelevant to practical applications [3]. If we are to reverse this trend, it is increasingly important for the CS curriculum to stay relevant to today's reality and engage students by making a strong connection between computing and their everyday lives. Introducing students to mobile applications and game development may help students to better relate to the course material and make stronger connections to real-world applications and gadgets they see and use every day.

Using computer games as a motivational and instructional tool in CS has recently become a popular subject. Introduction of game programming and game development courses or even entire degree programs has also received a wide attention. However, there are very few reports from educators about using mobile game development to increase student interest and engagement in pursuing a CS degree. This is surprising because mobile game programming is by far less complex than traditional game development due to the smaller scale of mobile games, simpler graphics, and other factors. Therefore, it may often be easier to adopt mobile game development in the CS curriculum than traditional game development.

In this paper, we present an approach that introduces mobile game development early in the CS curriculum. Along with programming a mobile game, we gently introduce students to many advanced CS topics, such as artificial intelligence, databases, and networking. By demonstrating these and other non-programming and diverse aspects of the discipline to the students, this approach may help dissolve a widely popular misconception that "CS is all about coding." Most CS students seem to be very interested in computer game development, and introducing students to this topic early in the curriculum could serve as a good tool to increase student retention.

2. BACKGROUND
Computer games have been successfully applied to improve recruitment and retention [1]; they have been shown to be a successful learning tool by leveraging students' enthusiasm towards computer games and their social relevance. Current students, more than at any time before, need a classroom experience that they can relate to, that is creative and challenging, and makes a difference in preparing for their careers. To address this challenge, design, programming, and development of computer games has been used in CS classrooms in many ways: as a general engagement tool [2,10]; as motivators and examples in introductory programming courses [1,17,28]; as capstone courses [1,24,29]; in interdisciplinary programs connecting CS

SIGCSE'09, March 3-7, 2009, Chattanooga, Tennessee, USA.

with arts and humanities [22,28,30]; as a well-fitting instrument to teach computer graphics [27,28], artificial intelligence [28,29], project management [30], and software engineering [8,13,28]; or as a concentration within existing CS programs [11,31]. A number of surveys indicate that majority of students prefer programming projects and assignments that one way or another involve computer games [9]. Furthermore, many students say that their choice of CS as a major was sparked by computer games [24,29].

So far, compared to integrating traditional computer game development into CS curricula, mobile games received relatively little attention. Mobile game development has been successfully integrated into graduate degrees [31]; they have been found especially helpful as a motivating tool to increase student interest in CS and other STEM disciplines [21]. Mobile devices have been integrated into CS curriculum in many ways, including teaching computer literacy [19] and introductory programming [20]. Current research indicates that mobile devices and mobile games are a valuable teaching and learning tool [7], especially in the context of attracting schoolchildren to computing [26] and as a powerful tool used in augmented learning [15]. Mobile game development also gives students the benefit of instant gratification – they can quickly build a working graphical application (that is more satisfying than calculating an average salary) and play the resulting game on their own mobile device [31].

Experience in mobile applications and game development may also serve as an additional motivational factor because it is a skill in high demand by the industry [4]. According to Gartner Research, the revenue in mobile gaming industry is expected to reach $4.5 billion in 2008, which represents a 16.1% growth compared to 2007. Gartner projects that the industry will grow at the rate of at least 10% per year through 2011. In the US, 36% of surveyed heads of households reported that they play games on their mobile devices, up from 20% in 2002 [12]. The target mobile phone market is very accessible and extremely large with estimates of over two billion phones in use worldwide.

Many critics of using computer games in the curriculum argue that they promote misogynistic and violent content [14] and that women are generally less interested in games than men [17]. Indeed, many console and PC games are marketed to young males [23]. However, in 2008, 40% of all game players in the US were women [12]. Furthermore, according to several surveys, women enjoy games-oriented introductory CS courses more than men [14, 17], and their average grades are often higher than men's [6].

Casual games [16] are the most popular kind of mobile games, mostly due to the patterns of mobile phone usage. Casual games are characterized by extremely simple gameplay (e.g. puzzles or card games) and generally appeal to "casual consumers" who do not regard themselves as "gamers". Casual games are typically played in short bursts: during work/class breaks, while waiting in line, and on public transportation. Their rules are simple, and, unlike many console games, they do not require a long-term time commitment or special skills. Most importantly, many studies indicate that mobile gamer demographics are split fairly equally between males and females [12, 16].

One of the challenges in adopting game-related coursework into CS curriculum is to balance the educational and motivational aspects by choosing the right mix of the material that students find interesting and the material that would help them grow academically [25]. As some critics of using games in CS classrooms noted, we should not focus on games simply because students like games [14]. Another substantial barrier to adopting game development into the curriculum is the inherent complexity of developing a playable and well-designed game [5]. Developing playable games is a complex task, but due to its complexity it also opens an opportunity to introduce students to many other advanced areas in CS. We believe that mobile game development addresses both of these challenges. Due to relative simplicity of many mobile games, a playable game can be developed within the timeframe of one semester by many CS students with reasonable Java programming skills. As we show in this paper, mobile games can also be successfully used to broaden the horizons of students in introductory CS courses by exposing them to a variety of advanced topics early in the curriculum.

3. MOBILE GAME DEVELOPMENT

In this paper we describe an approach to teaching a Mobile Game Development course that is positioned early in the CS curriculum. The only required prerequisite is CS1: students must have some programming experience in Java. This course is based on a semester-long project to develop a playable game supplemented with an assortment of smaller assignments. Although we offer this course to all students meeting the prerequisite, it is ideally suited for freshmen with Advanced Placement (AP) CS credit for whom it will be their first college-level CS course.

The following sections emphasize the main topics discussed in our course. For the most part, they are presented here in the same sequence as in the course. However, different aspects of several topics, including data structures and software engineering, are discussed throughout the course.

3.1 Programming Mobile Devices

We begin by introducing student to the architecture of mobile devices and the features that set them apart from desktop computers or gaming consoles. It is important for students to understand that small screens, limited CPU and battery power have a significant impact on the types and features of applications that can run efficiently on mobile devices. Students also appreciate the fact that most of their mobile phones have as much computing power as many PCs manufactured 10-15 years ago.

A number of software platforms are available for developing software running on current mobile devices. These include Sun's Java 2 Platform Micro Edition (J2ME), Qualcomm's BREW, Microsoft Windows Mobile, Symbian, and Palm OS. We chose J2ME because it is widely adopted in CS1 courses and because it as a required element in the AP CS courses. Sun Java Wireless Toolkit (WTK, http://java.sun.com/products/sjwtoolkit) provides a flexible environment for testing, debugging, and deploying J2ME-based mobile applications via a customizable emulator available for Windows and Linux. J2ME provides a number of packages within its MIDP (Mobile Information Device Profile) API designed especially for mobile game development.

3.2 Computer Graphics

We believe that it is very important to introduce more engaging elements early in the course before students have a chance to lose their interest. Computer graphics is a fundamental element in any computer game and we begin by discussing such necessary

prerequisites as graphics coordinate system and drawing graphics primitives. In this course, we focus on 2D graphics and animations, which are supported by MIDP API and its `GameCanvas` and `Sprite` classes. When discussing animation, it is essential that students have a firm grasp on the concept of frame rate and how computer animation relates to traditional animation. Good understanding of these concepts helps introduce collision detection. MIDP API supports several types of collision detection. Additionally, `GameCanvas` class is provides special tools to achieve smooth animation using double buffering.

Given the limited computational capabilities and small screens of mobile devices, it may be difficult to practice any advanced computer graphics. However, this context presents an excellent opportunity to talk about such topics as image processing and sampling, alpha blending, 3D rendering, etc.

3.3 HCI and Multimedia

We introduce students to basic design elements and features of mobile and casual games. Interaction modalities used in computer games are very different from those found in other types of software. Mobile applications need to take into account a limited set of interaction options, such as tiny keypads and small screens. Students get exposed to basic research elements of human-computer interaction (HCI) as they make a connection between the usage patterns of mobile devices and the types of games and their narratives that may be suitable for the limited attention span of a typical mobile user. We also discuss design patterns common for different types and genres of console and mobile games.

Animation plays an extremely important role in computer games. Since students have already been introduced to technical aspects of creating animation form mobile games, we discuss computer animation, music, and sound effects as main driving tools used in creating engaging mobile games. Students usually enjoy a number of exercises offered to them that involve programming mobile devices to play tonal, WAVE and MIDI sounds. In particular, we received especially positive feedback regarding the exercise that involves creating a small on-screen piano keyboard that can be played using the mobile device's numeric keypad.

3.4 Data Structures and Algorithms

Tiled images used to create screens and backgrounds in the game are often organized as two-dimensional arrays, which is a good point to start a discussion about other types of data structures that may be implicitly used in a mobile game. For example, queues can be introduced using a situation when a mobile user could press keys faster than they could be processed by a slow CPU. Multiple screens comprising a larger game landscape could be used to introduce graphs. Naturally, this may lead to discussions about algorithms that are better suited for processing each of the encountered data structures. Although many of them are implemented by J2ME framework, students often have a chance to practice their own ideas. For example, in a game involving many characters simultaneously moving on the screen, students get to decide for themselves how and in which order they will perform collision detection between the pairs of corresponding sprites. This leads to a discussion about the efficiency of different types of algorithms because students often can see firsthand how an inefficient algorithm can ruin a game playing experience by making it too slow.

3.5 Artificial Intelligence

The vast majority of casual mobile games are designed for a single player. Many game types, e.g. logical puzzles, typically act only in response to the user actions. For example, in Bejeweled, the game logic removes the line of adjacent gems of the same color formed by the user, and then allows the gems above to fall down to fill the gaps. Other types of games may involve more sophisticated AI-controlled behaviors making it possible for the user to play against the mobile device. AI techniques used in game development include rule-based character behavior (e.g. chasing/evading), adversarial search in turn-based games such as checkers. We ask students to implement several behavior strategies to manipulate different game characters, in which game-controlled characters will chase or evade the main character controlled by the user. Several examples are shown in Figure 1.

3.6 Database Management

Students in our Mobile Game Development class were often proud of their achievements and some liked to show off their projects to friends. High scores persistently stored on a mobile device could only add more bragging rights to the developer and the player of the game. This naturally leads to the discussion of the difference between persistent and. volatile data storage. J2ME offers `RecordStore` class, which implements a very simplistic two-column database table addressable by a unique numeric identifier. Illustrating the basic operations of this class such as record storage and retrieval, presents a good starting point for a discussion on the subject of database management.

3.7 Computer Networking

Once students gain some experience in implementing single-player mobile games, they begin asking questions about multiplayer mobile games. While this serves as an excellent starting point to begin the discussion about computer networking, we had mixed results with related hands-on activities. One reason is that this topic usually comes up towards the end of the course and there may not be a sufficient amount of time to add multiplayer features to existing projects or implementing a new multiplayer game from scratch. Another reason is that of all other advanced topics reviewed in this course, implementing data communication in J2ME requires comparatively higher technical skills than those possessed by students in this class. However, students always enjoy and usually have enough skills to experiment using WTK with the existing code that implements a client/server communication between two mobile devices.

3.8 Software Engineering

Design and development of computer games serves as an excellent framework for teaching software engineering [8]. We believe that good software engineering principles should be presented and applied throughout the CS curriculum, which is a practice that we also apply in this course that is based on a semester-long project that emphasizes teamwork. Throughout the course, we gently introduce the principles of software quality, present many examples of refactoring, and stress code optimization which plays an increasingly important role on mobile devices. At the end of the course, we dedicate some time to review all these and other practices introduced throughout the course and frame them within the software engineering discipline.

Figure 1. Mobile games created by students in the Mobile Game Development course: Snack Runner (left), Rally (middle), Mobile Doom (right)

4. STUDENT FEEBACK

This course framework has been offered to undergraduate students several times and is constantly evolving. The unifying theme stays the same: to engage students early in the CS curriculum through highly motivating applications using mobile devices. One of our goals was to convince the students who decided to give CS a try that "it's not all about coding" and that it includes many other diverse areas. Many students agreed:

- *The course definitely sparked my interest in [other areas] and made me want to explore.*
- *It gave me more experience and depth of knowledge in certain areas.*

Students liked the fact that in this course they were exposed to several advanced topics. Most importantly, this motivated some students to learn more about them:

- *It did make me want to learn about databases, networking, etc. to broaden my understanding of CS. Not be limited to just the software side.*
- *I enjoyed learning about new topics such as AI. Our games [were] playable over the network, so now I am more interested in networking.*
- *I think this course [exposed] the software engineering part of CS. I think it gave me more of an idea as to how much work and time goes into making a game.*

Many students appreciated the teamwork and the semester-long project that resulted in creating playable mobile games (Figure 1):

- *The project was the most fulfilling and enjoyable part of the course.*
- *It's always good to be exposed to group work -- few projects in the real world are done by a single individual or no input from others. And we all know that us CS folk could always use some more practice dealing with others.*
- *I liked how everyone got to make their own game.*

5. CONCLUSION

Computer games gained a wide acceptance as an engaging and motivating tool in the CS curriculum. However, designing and implementing a playable game is a very challenging task and is best implemented in advanced courses where students already have a sufficient experience in software development and exposure to other CS topics. Mobile games offer an advantage of being simpler by nature and thus easier to program. This makes it more feasible to lower-level students to develop playable games as a part of their classroom experience.

Mobile applications are often easy for students to relate to, because mobile technology plays an increasingly important role in the lives of today's students. For many of them, their mobile phone is replacing a desktop computer as their primary computing device. Mobile applications and games offer instant gratification in the sense that students can download them to their mobile phones almost immediately and show them off to their friends.

In our approach, we strive to make connections to advanced CS topics as early in the curriculum as possible. Many studies indicate that students often get disillusioned in CS because they see too many irrelevant or contrived examples that are boring or have very little to with real-world applications. There is a certain percentage of students who are interested in other areas of CS that emphasize design or theory over coding. By exposing students to a wide range of advanced CS topics early in their academic career, our approach aims to show students that CS can be much more exciting than coding and that there are so many areas in which programming plays a supplementary role. Course structure described in this paper presents a cohesive and engaging theme for introducing many advanced topics early in the CS curriculum.

6. REFERENCES

[1] Barnes, T., Richter, H., Powell, E., Chaffin, A., and Godwin, A. 2007. Game2Learn: building CS1 learning games for retention. SIGCSE Bull. 39, 3 (Jun. 2007), 121-125.

[2] Bayliss, J. D. and Strout, S. 2006. Games as a "flavor" of CS1. SIGCSE Bull. 38, 1 (Mar. 2006), 500-504.

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

[4] Brandel, M. 2007. 12 IT skills employers cannot say no to. ComputerWorld, Jul. 2007.

[5] Burd, B., Goulden, J., Ladd, B., Rogers, M., and Stewart, K. 2007. Computer games in the classroom, or, how to get perfect attendance, even at 8 AM. SIGCSE Bull. 39, 1 (Mar. 2007), 496-496.

[6] Chamillard, A. T. 2006. Introductory game creation: no programming required. SIGCSE Bull. 38, 1 (Mar. 2006), 515-519.

[7] Chao, C. 2006. An Investigation of Learning Style Differences and Attitudes toward Digital Game-based Learning among Mobile Users. In Proceedings of the 4th IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education (Nov. 16-17, 2006). WMTE. IEEE CS, Washington, DC, 29-31.

[8] Claypool, K. and Claypool, M. 2005. Teaching software engineering through game design. SIGCSE Bull. 37, 3 (Sep. 2005), 123-127.

[9] Cliburn, D. C. and Miller, S. 2008. Games, stories, or something more traditional: the types of assignments college students prefer. SIGCSE Bull. 40, 1 (Feb. 2008), 138-142.

[10] Clua, E., Feijó, B., Rocca, J., Schwartz, J., das Graças, M., Perlin, K., Tori, R., and Barnes, T. 2006. Game and interactivity in computer science education. In ACM SIGGRAPH 2006 Educators Program (Boston, MA, Jul. 30 - Aug. 3, 2006). SIGGRAPH '06. ACM, New York, NY.

[11] Coleman, R., Krembs, M., Labouseur, A., and Weir, J. 2005. Game design & programming concentration within the computer science curriculum. SIGCSE Bull. 37, 1 (Feb. 2005), 545-550.

[12] Entertainment Software Association. 2008. Essential facts about the computer and video game industry. http://www.theesa.com/facts/pdfs/ESA_EF_2008.pdf.

[13] Gestwicki, P. V. 2007. Computer games as motivation for design patterns. SIGCSE Bull. 39, 1 (Mar. 2007), 233-237.

[14] Haller, S., Ladd, B., Leutenegger, S., Nordlinger, J., Paul, J., Walker, H., and Zander, C. 2008. Games: good/evil. SIGCSE Bull. 40, 1 (Feb. 2008), 219-220.

[15] Klopfer, E. 2008. Augmented learning: research and design of mobile educational games. MIT Press.

[16] Koivisto, E. M. 2006. Mobile games 2010. In Proceedings of the 2006 international Conference on Game Research and Development (Perth, Australia, Dec. 4-6, 2006). ACM International Conference Proceeding Series, vol. 223, 1-2.

[17] Leutenegger, S. and Edgington, J. 2007. A games first approach to teaching introductory programming. SIGCSE Bull. 39, 1 (Mar. 2007), 115-118.

[18] Lewis, M. C. and Massingill, B. 2006. Graphical game development in CS2: a flexible infrastructure for a semester long project. SIGCSE Bull. 38, 1 (Mar. 2006), 505-509.

[19] Mahmoud, Q. H. and Dyer, A. 2007. Integrating BlackBerry wireless devices into computer programming and literacy courses. In Proceedings of the 45th Annual Southeast Regional Conference (Winston-Salem, NC, Mar. 23-24, 2007). ACM-SE 45. ACM, New York, NY, 495-500.

[20] Mahmoud, Q. H., Dyer, A. 2008. Mobile Devices in an Introductory Programming Course. Computer, 41, 6, (Jun. 2008), 106-108.

[21] Metcalf, D., Milrad, M., Cheek, D., Raasch, S., Hamilton, A. 2008. My Sports Pulse: Increasing Student Interest in STEM Disciplines through Sports Themes, Games and Mobile Technologies. In Proceedings of the 5th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education (Beijing, China, Mar. 23-26, 2008). IEEE CS, Washington, DC, 23-30.

[22] Murray, J., Bogost, I., Mateas, M., and Nitsche, M. 2006. Game Design Education: Integrating Computation and Culture. Computer 39, 6 (Jun. 2006), 43-51.

[23] Natale, M. J. 2002. The effect of a male-oriented computer gaming culture on careers in the computer industry. SIGCAS Comput. Soc. 32, 2 (Jun. 2002), 24-31.

[24] Parberry, I., Roden, T., and Kazemzadeh, M. B. 2005. Experience with an industry-driven capstone course on game programming: extended abstract. SIGCSE Bull. 37, 1 (Feb. 2005), 91-95.

[25] Repenning, A. and Ioannidou, A. 2008. Broadening participation through scalable game design. SIGCSE Bull. 40, 1 (Feb. 2008), 305-309.

[26] Sharples, M. 2000. The design of personal mobile technologies for lifelong learning. Comput. Educ. 34, 3-4 (Apr. 2000), 177-193.

[27] Sung, K., Shirley, P., and Rosenberg, B. R. 2007. Experiencing aspects of games programming in an introductory computer graphics class. SIGCSE Bull. 39, 1 (Mar. 2007), 249-253.

[28] Sweedyk, E., deLaet, M., Slattery, M. C., and Kuffner, J. 2005. Computer games and CS education: why and how. SIGCSE Bull. 37, 1 (Feb. 2005), 256-257.

[29] Wallace, S. A., Russell, I., Markov, Z. 2008. Integrating Games and Machine Learning in the Undergraduate Computer Science Classroom. In GDCSE 2008: Microsoft Academic Days Conference on Game Development in Computer Science, (Celebrity Century, Feb. 28 - Mar. 2, 2008).

[30] Wolz, U. and Pulimood, S. M. 2007. An integrated approach to project management through classic CS III and video game development. SIGCSE Bull. 39, 1 (Mar. 2007), 322-326.

[31] Zyda, M. J., Thukral, D. Ferrans, J. 2008. Incorporating Mobile Games into a Computer Science Game Degree Program. In GDCSE 2008: Microsoft Academic Days Conference on Game Development in Computer Science, (Celebrity Century, Feb. 28 - Mar. 2, 2008).