What Students Think About Game-Themed CS1

Antti-Jussi Lakanen and Vesa Lappalainen. University of Jyväskylä, Finland

Abstract

There has been rising academic interest in including computer game programming in CS1 curriculum during the past years, and many experience reports exist on the topic. Still, there is a need for knowledge on student reception, that is, how the students themselves view the game theme on a CS1 course. In this paper we review our CS1 course feedback from the years 2008–2013, of which three first years were carried out without games, while the latter three years part of our weekly assignments, plus a bigger course assignment, have been "game-themed". We found that over 90 % found contextualization through games fun, motivating or rewarding in some other way. Less than 10 % of the respondents have somewhat a negative overtone about the course theme. At the same time, dropout rate has declined more than 12 percentage points.

CR Categories: K.3.2. [Computers and education]: Computers and Information Science Education—[Computer Science Education]

Keywords: CS1; game programming; game theme; motivation

1 Introduction

Partly prompted by a severe drop in CS major applicants some years ago, a more contextualized and contemporary method of teaching and learning computer science has been called for [Leutenegger and Edgington 2007]. Academic interest for adopting computer games has been increasing in the past few years – more and more instructors and institutions have altered their curricula to include more games in some form. In computer science education, computer games have been used to relate abstract principles to real-world experience. As a result, instructors hope their courses to be more appealing to students. Playing ("consuming") computer and mobile games has become a part of everyday life for young people, and games are indeed seen as a fruitful way of bringing computing education into context. Firstly, they incorporate many disciplines such as mathematics, artificial intelligence, and physics, as well as art and design, serving as a catalyst for arousing motivation for learning CS [Guzdial and Soloway 2002]. Secondly, regarding the human learning process, it is important to provide meaning and motivation for learners [DeClue 2009; Guzdial 2010].

Despite the growing popularity of incorporating games into CS curriculum, there exists surprisingly little research on student reception, and therefore more research on this area is needed. In this article, we aim to develop an understanding of students' views and opinions of what we call a "Game-Themed CS1". The study is twofold. First, we will address the student performance based on weekly activity and exam scores, and present comparisons with the course instances carried out without a game theme. Second, we focus on free-text survey responses from three CS1 courses during a three-year period. The valid number of survey respondents was 269. We then construct descriptive categories of student responses through qualitative data analysis in which we address the second dataset discussed above.

2 Motivating CS1 through programming games

It is important to acknowledge that games (or play) in a broader sense are most likely to be part of every CS1 course, for example, in textual "games", like tossing coin, guessing a number, or Hangman. Later in this article, the *game theme* specifically refers to integrating interactive, graphical computer games into CS1 curriculum.

The literature indicates that games can attract both males and females [Leutenegger and Edgington 2007; Luxton-Reilly and Denny 2009], and add to students' confidence with regard to their learning abilities [Leutenegger and Edgington 2007]. Graphical games have been argued to match well with the constructivist view of learning, as the visual experimentation involved is likely to help in developing internal models of programming concepts [Luxton-Reilly and Denny 2009]. Games are also socially relevant, as students can share their games within their own network [Rajaravivarma 2005].

There are many ways to integrate computer games into CS classes, as reviewed by Sung [2009]. In the present study, the focus is in what Sung calls a "game development client", which refers to existing CS classes that "creatively integrate games into their existing curriculum." However, as an existing CS1 course is typically not a game development class *per se*, it might be challenging to find a balance between reaching learning objectives and at the same time making games attractive enough. The difficulty of implementing attractive games has been regarded as the key challenge of game development in introductory programming courses [Giguette 2003]. We find that this challenge has been mitigated by tooling the game development properly [Isomöttönen et al. 2011; Leutenegger and Edgington 2007], by letting the students start with a pre-programmed skeleton [Luxton-Reilly and Denny 2009], and by emphasizing games that are of tolerable size [Kurkovsky 2009].

We have previously studied university students' acceptance of our game-themed CS1 course [Isomöttönen and Lappalainen 2012]. The present study complements and extends this earlier work.

3 Our CS1

Our CS1 course (6 ECTS, 160 hours) uses procedural paradigm – with using ready-made objects from libraries. $C\#$ serves as our first language. Course completion involves passing an exam, a sufficient number of completed tasks during each week, and a course assignment. From the learning objectives point of view our course consists of typical CS1 concepts [Dale 2005], like selection, repetition, information encapsulation, and so on. As a technical framework for game development we use *Jypeli*[1] programming library, which is described in detail in [Isomöttönen et al. 2011]. Jypeli is built on top of XNA framework, and we think that the primary function of Jypeli is that it "lowers the barriers to programming" (see, [Kelleher and Pausch 2005]), while XNA alone is not especially

[1] `http://bit.ly/jypeli-en`

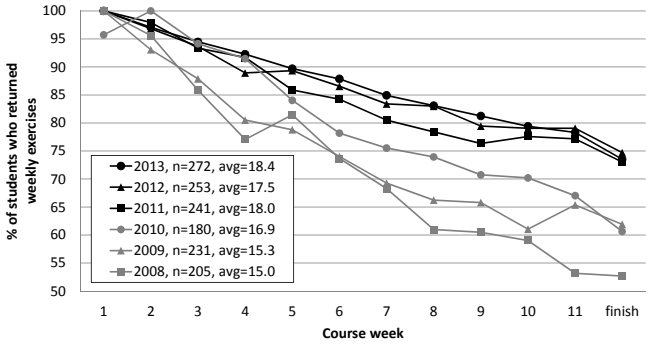

Figure 1: *Comparison of student performance in 2008–2013. The black curves represent course instances with game theme, while the gray curves represent courses without games. Number n refers to the number of students on the first week, and avg refers to the average exam score (max 24). Note, that all the instances are held by the same lecturer.*

beginner friendly. We have used [Lakanen et al. 2014] the Jypeli library also with our middle and high school outreach activities. Due to the fact that we quite heavily lean on the library, we also mention – if not emphasize – some central object-oriented topics, like determining classes, responsibilities and inheritance.

Regarding the game theme, each week some of the weekly tasks relate to game programming. Right from the first week, the students get to do graphical programs, draw shapes with different colors, make them move with realistic physics, etc. Students are kept away from writing boilerplate code, which keeps them more focused on the game content and learning objectives. Towards the end of the course, each student works with a bigger course assignment (30 work hours) as they make a working computer game. Typically, the games can include multiple players, inputs from different devices, creating levels from string arrays, gauges and other indicators, and so on. Examples of student games can be found at `http://goo.gl/OGJKFB`.

Due to integrating the game theme, there were some changes also to the course model. First, with the game theme, figuring out own game ideas (player controls, game mechanics, scoring systems, etc.) and making the game designs now starts very early on the course. Altogether there are three checkpoints to support the development process. Regarding the earlier model, the students mostly chose their assignment from a ready-made list of "traditional" subjects. They also started working with the assignment very late, basically centering around the final weeks of the course.

4 Student performance

We first piloted the game theme in CS1 back in 2010 with CS majors only. After the encouraging experiences and feedback the game theme was widely adopted for all students, majors and minors, in 2011. Since then, all our CS1 courses have been game-themed. In the Figure 1 we present comparison of student performance between the game-themed courses in 2011–2013 with the courses *without* specifically concentrating on games in 2008–2010. The figure illustrates the students' activity in returning the weekly assignments. The number of week is on the x-axis, and the relative proportion of the students who returned assignments on the y-axis.

During the last three years, when the games have been a substantial part of the course, we see a rise in the relative amount of students returning assignments, and reduction in the dropout rate. While in 2008 the dropout peaked 47.3 % *without* the game theme, in 2012 the dropout was 25.7 % *with* the games. The amount of students taking the course has also steadily risen from 2008 (108 passed students) to 2013 (199 passed students).

There are a few considerations to be made on the comparison based on the figure presented above. First, this comparison does not necessarily imply better learning results — the final exam scores have not declined but rather inclined a little — however, this is a notion to be further investigated later by clustering students according to their course performance, and also comparing the exam questions. Secondly, along with introducing the game theme, we made some additional adjustments based on our earlier research. For example, we introduced a model for reinforced student support [Haatainen et al. 2013]. The objective of the reinforced support model is to create a comfortable study environment, to lower the social and mental obstacles of active participation, and to encourage students to ask questions.

Third, in 2009 and 2010 the curves have somewhat climbed from the numbers in 2008. This may be due to the adopting "Media computation" (see, [Guzdial 2003; Simon et al. 2010]) as a part of the CS1 curriculum and revising the assignments accordingly.

5 Themes extracted

The main focus of the present study was to get a holistic view of how students experience the game theme in a CS1 course. Getting as complete picture as possible is important, since the game theme (and contextualization at length) had such a big role in the course. Therefore, we treated dataset as a whole — we wanted to get the "big picture", to see students both as individuals and as a cohort.

As for the data analysis procedure, we carefully examined the survey responses concerning specifically the game theme. We interpreted an initial revision of the themes that characterise notions of the original data. Then, we iteratively reviewed our data several times, and along with several adjustments made to the coding scheme, elaborated the emerging themes. The analysis process resembles pattern coding in qualitative research, where regularities in the data are identified and finally low level themes form into higher level categorial scheme [Miles and Huberman 1984, pp. 67–69]. The first author made the initial categorization, after which the second author presented questions and pointed out his considerations. These observations were taken into account in the later iterations of the analysis.

Altogether, five distinct themes were identified. Theme I, *Fun, motivating*, was clearly the dominant theme. Another theme with a positive overtone was Theme II, *Creativity*. Neutral themes regarding the games were Theme III, IV, and V, *Does not matter*, *Burdensome*, and *Other*. The themes with somewhat a negative or unfavourable nuance were Themes VI and VII, *Does not teach enough*, and *Not interested in games*.

5.1 Fun, motivating

A clear majority of the students found the game theme to be a very appropriate contextualization for a CS1 course. Making a complete game him-/herself gave a bigger picture of what it takes to develop a fully working computer program, and especially creating a working game was rewarding. This is illustrated in the quote below.

> Making an entire game illustrated me the process of making [a complete] computer program.

The *fun* was also a dominant overtone in the respones, and many of the responses is phrased "There were challenges – – but it was fun." It seems that with games a more genuine interest toward computer

programming emerged, and students were no longer only making just a mandatory assignment, but were also ready to exceed initially conceived task demands in their work (cf. [Hidi and Renninger 2006]).

> I had that much fun – – that I could hardly stop. When I got one feature ready, I came up yet another one, which I wanted to implement. You can't necessarily say the same about other courses' assignments.

5.2 Creativity

Game theme enabled the students to focus more on the content and the creative side of game development process, and the *handprint* of student himself is concretely visible. Naturally, students had a possibility to design the game characters, which actually was the starting point for many. The possibility of seeing visually the results made it possible to recognize problematic areas in the game.

> Very good theme. I was able to apply what I had learned, which motivated me, because it make the result feel "my own" and important.

There were several comments, that made comparisons between a theme that the students had found "interesting" and "just making some boring applications". One important aspect, that we will focus more on the Section 5.5, is it was not a part of the learning objectives to go through the details of the library. However, despite that the library "masked" some implemental details that students may have liked to understand better (say, how the physics work under the hood), this was generally not a problem in this category, which is illustrated in the student quote below.

> Not all the musicians do understand the theories behind smash hits either, but still they are able to create great songs.

5.3 Does not matter

One distinct theme – only showing in few responses, though – emerging from the surveys, was that the theme or contextualization seemed irrelevant, but did not hinder anything, on the other hand. Students were pleased on the educational experience, but did not stress the game theme specifically.

5.4 Burdensome

A theme appearing every now and then was that making a working computer game or game theme in more general was laborous. One student commented, that games would have to be really simple to be able to finish them in the time budgeted for making them.

Generally, the responses falling in this category are neutral regarding the game theme. Compared to the earlier course model, we assume that the amount of work has not been decreased – rather the opposite. Still, our experience shows that students are ready to seek information on the library references in a self-directed manner.

5.5 Does not teach enough / want to start from atoms

This was the second most frequent theme emerging from the analysis and probably deserves the most reflection of all the themes discovered. The most important argument found was that the game theme is not the best alternative for educational purposes, because it gives only a shallow contact with the programming. Going the "traditional way" (without graphical games, we suppose) would have been more effective and valuable at root, because there exists a number of "more conventional fields, where [programmers] are needed." This suggests, that the link between the course topics and applicability to other courses or working life was not crystal clear through the game theme.

Respondents also suggested, that making games does probably not yield good learning results, because the game library does that much for the programmer, and game related assignments can be done mainly with using the library methods without really understanding the details of the solution.

> It is true, that through the games we are able to make [visible elements] to the computer display, and that's why it probably increases some interest. However, in my mind [the games] just confused me more, when we were trying to learn some basic programming.

For some not knowing or completely understanding things happening behind the curtain was very annoying.

> Personally, I do not like the magic happening in the background without understanding what is happening.

We have noted that in practice this shows in the assignment check situations, where it is sometimes difficult for the students to figure out where to "naturally" utilize loops, arrays or lists, as the library many times does the job for the programmer.

Even though this category was marked as "negative", not all the responses were solely negative in tone.

> Excellent idea, but if the game should be done without Jypeli, it would probably be out of the question.

In the quote above, the student may have had an image that he should be able to develop games without the help of any library, and therefore he may be under the impression that he has not actually reached the learning objectives of the course. This suggests, that teachers should communicate more clearly that it is not necessary to understand in detail how the library works under the hood, and that it is important to instruct the students to focus on the aspects that are central to the learning process.

5.6 Not interested in games

During the initial stages of our "theming" transition we were somewhat worried about that there would be quite many students who would not like games at all. As it turned out, a surprisingly low number of responses were categorized under this theme. Yet, we must not dismiss the theme, and in fact in the beginning of each course we have emphasized that it is possible to *not* to choose the bigger course assignment to be a computer game. Also, in the weekly assignments there are many tasks to choose from, which are not directly related to games, and is it possible to reach the weekly "maximum score" without game-themed assignments.

The students who expressed categorically not being interested in games, often added something about their age or maturity. An example of a student quote is below.

> A geezer aged 45+ does not feel like becoming interested in games. Something more close to business world.

For some the games were not bad *per se*, but would still have wanted something else, as did the student below, to whom computer science was a minor subject.

> The game theme seemed a "bread and circuses" type of solution – – As a more science inclined student I was left with an impression that the course was much of the topping, but only a little of the cake.

Nuance	Theme	f	%
Positive	Fun, motivating	152	74.5
	Creativity	10	4.9
			79.4
Neutral	Does not matter	11	5.4
	Burdensome	5	2.5
	Other	3	1.5
			9.4
Negative	Does not teach enough	14	6.9
	Not interested in games	9	4.4
			11.3
Total		**204**	

Table 1: *Themes extracted with frequencies.*

6 Conclusion

In this paper we reviewed students' views and opinions about a game-themed CS1. We found that a clear majority finds the game theme itself fun, motivating and beneficial in different ways. For example, games-related assignments were experienced concrete, creative and worth of putting many hours of work in them. Less than 10 % of the respondents viewed them uninteresting, too burdensome, or a bad educational choice. The 90% / 10% distribution was practically the same when compared to the differently worded Likert-scale questions in the same survey. Still, 10 % is a big number and has to taken into account in the future courses with communicating more clearly the different alternatives or categories for the assignments that the students can choose from.

In addition to the positive feedback, the dropout rate descended from 47 % to 26 %, though there may be some additional aspects that have to be consider to be part of the reduced dropout.

Regarding the future research, we will look into at least the following aspects. First, we have run the course with the presented contextualization now for a few times. How it affects the student views on the following courses, like CS2 and so on, is yet to be investigated. Is there a risk, that if students get use to a really "fun way", they might experience the "traditional way" of CS2 or later courses boring or unattractive? At least for now this phenomena is not in sight at large, but it is important for the teachers to acknowledge the risk.

Secondly, the actual learning results and long-term implications would be a good area of research. Does the game theme imply better learning results? As we presented in the Section 4 the dropout rate has declined during the past three years, however, the same phenomena is not present in the CS2 courses.

References

DALE, N. 2005. Content and emphasis in CS1. *SIGCSE Bull. 37*, 4, 69–73.

DECLUE, T. 2009. A theory of attrition in computer science education which explores the effect of learning theory, gender, and context. *J. Comput. Sci. Coll. 24*, 5 (May), 115–121.

GIGUETTE, R. 2003. Pre-Games: Games designed to introduce CS1 and CS2 programming assignments. In *Proceedings of the 34th SIGCSE technical symposium on Computer science education*, ACM, New York, NY, SIGCSE '03, 288–292.

GUZDIAL, M., AND SOLOWAY, E. 2002. Teaching the Nintendo generation to program. *Commun. ACM 45*, 4 (Apr.), 17–21.

GUZDIAL, M. 2003. A media computation course for non-majors. *SIGCSE Bull. 35*, 3 (June), 104–108.

GUZDIAL, M. 2010. Does contextualized computing education help? *ACM Inroads 1*, 4 (Dec.), 4–6.

HAATAINEN, S., LAKANEN, A. J., ISOMOTTONEN, V., AND LAPPALAINEN, V. 2013. A practice for providing additional support in CS1. In *Learning and Teaching in Computing and Engineering (LaTiCE), 2013*, IEEE, 178–183.

HIDI, S., AND RENNINGER, K. A. 2006. The Four-Phase Model of Interest Development. *Educational Psychologist 41*, 2 (June), 111–127.

ISOMÖTTÖNEN, V., AND LAPPALAINEN, V. 2012. CS1 with games and an emphasis on TDD and unit testing: piling a trend upon a trend. *ACM Inroads 3*, 3 (Sept.), 62–68.

ISOMÖTTÖNEN, V., LAKANEN, A.-J., AND LAPPALAINEN, V. 2011. K-12 game programming course concept using textual programming. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, ACM, New York, NY, USA, SIGCSE '11, 459–464.

KELLEHER, C., AND PAUSCH, R. 2005. Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. *ACM Comput. Surv. 37*, 2, 83–137.

KURKOVSKY, S. 2009. Engaging students through mobile game development. *SIGCSE Bull. 41* (March), 44–48.

LAKANEN, A.-J., ISOMÖTTÖNEN, V., AND LAPPALAINEN, V. 2014. Five years of game programming outreach: Understanding student differences. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, ACM, New York, NY, USA, SIGCSE '14, 647–652.

LEUTENEGGER, S., AND EDGINGTON, J. 2007. A games first approach to teaching introductory programming. *SIGCSE Bull. 39*, 1, 115–118.

LUXTON-REILLY, A., AND DENNY, P. 2009. A simple framework for interactive games in CS1. *SIGCSE Bull. 41* (March), 216–220.

MILES, M., AND HUBERMAN, A. 1984. *Qualitative data analysis: a sourcebook of new methods*. Sage Library of Social Research. Sage Publications.

RAJARAVIVARMA, R. 2005. A games-based approach for teaching the introductory programming course. *SIGCSE Bull. 37* (December), 98–102.

SIMON, B., KINNUNEN, P., PORTER, L., AND ZAZKIS, D. 2010. Experience report: CS1 for majors with media computation. In *ITiCSE '10: Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, ACM, New York, NY, USA, 214–218.

SUNG, K. 2009. Computer games and traditional CS courses. *Commun. ACM 52*, 12, 74–78.