

A Gameful Approach to Teaching Software Design and Software Testing — Assignments and Quests

Swapneel Sheth, Jonathan Bell, Gail Kaiser
Department of Computer Science, Columbia University, New York, NY, USA
{*swapneel, jbell, kaiser*}@cs.columbia.edu

1. Introduction

Introductory CS classes typically do not focus on software testing [5,6]. A lot of students' mental model when they start learning programming is that "if it compiles and runs without crashing, it must work fine." Despite numerous attempts to introduce testing early in CS programs and many known benefits to inculcating good testing habits early in one's programming life [4, 6], students remain averse to software testing as there is low student interest in software testing [5].

To address this problem, we used an internally developed research system called HALO — "Highly Addictive socialLy Optimized Software Engineering" [1]. Our previous work describes early prototypes of HALO; in this paper, we describe how we used it for the CS2 class and the feedback from real users. HALO uses game-like elements and motifs from popular games like World of Warcraft [2] to make the whole software engineering process and in particular, the software testing process, more engaging and social. HALO is not a game; it leverages game mechanics and applies them to the software development process. For example, in HALO, students are given a number of "quests" that they need to complete. These quests are used to disguise standard software testing techniques like white and black box testing, unit testing, and boundary value analysis. Upon completing these quests, the students get social rewards in the form of achievements, titles, and experience points. They can see how they are doing compared to other students in the class. While the students think that they are competing just for points and achievements, the primary benefit of such a system is that the students' code gets tested a lot better than it normally would have.

2. Background

COMS 1007, Object Oriented Programming and Design with Java, is the second course in the track for CS majors and minors at Columbia University. The first author taught this course in Spring (January-May) 2012¹. The course goals are "A rigorous treatment of object-oriented concepts using Java as an example language" and "Development of sound programming and design skills, problem solving and modeling of real world problems from science, engineering, and economics using the object-oriented paradigm" [3]. The prerequisite for the course is familiarity with programming and Java (demonstrated through a successful completion of the CS1 course at Columbia or another university, or passing marks on the AP Computer Science Exam).

In Spring 2012 the class enrollment was 129, which consisted largely of freshmen and sophomores (first and second year undergraduates). The list of topics covered were Object Oriented Design, Design Patterns, Interfaces, Graphics Programming, Inheritance and Abstract Classes, Networking, and Multithreading and Synchronization. There were five roughly biweekly assignments, which contained both theory and programming, and one midterm, and one final exam.

¹The introductory sequence of courses has undergone a change and COMS 1007 has become an honors version of the CS1 course since Fall 2012.

3. HALO in the classroom

In this class, we used HALO for three assignments. In the first two cases, HALO was not a required part of the assignment; students could optionally use it if they wanted to. For the last case, students could earn extra credit (10 points for the assignment, accounting for 0.8% of the overall course grade) by completing the HALO quests.

3.1. Assignment 2 and quests

3.2. Assignment 3 and quests

An Assignment on Java Networking — Getting and Analyzing Data from the Internet - The CIA World Factbook We now describe an assignment that was given to the class and the HALO quests that were created for it.

The CIA has an excellent collection of detailed information about each country in the world, called the CIA World Factbook. For this assignment, students had to write a program in Java to analyze data from the CIA World Factbook website, interacting directly with the website. The student programs had to interactively answer questions such as: 1. List countries in *South America* that are prone to *earthquakes*. 2. Find the country with the lowest elevation point in *Europe*. 3. List all countries in the *southeastern* hemisphere. 4. List countries in *Asia* with more than 10 political parties. 5. Find all countries that have the color *blue* in their flag. 6. Find the top 5 countries with the highest electricity consumption per capita. (Electricity consumption % population) 7. There are certain countries that are entirely landlocked by a single country. Find these countries. Note: For the italicized parts in the above, the code had to be able to deal with any similar input (e.g., from a user). This should not be hard coded.

HALO Quests We now describe the HALO quests that we used for the above assignment.

- i) **TARDIS** — To interact with the CIA World Factbook, it would be nice to have a TARDIS. No, not like in the show, but a java program that can Transfer And Read Data from Internet Sites. Completing this quest will reward you with 30 XP. This quest has two tasks: 1) *New Earth* — This will probably be your first program that talks to the Internet. While this isn't as complex as creating a new Earth, you should test out the basic functionality to make sure it works. Can you program read one page correctly? Can it read multiple pages? Can it read all of them? 2) *The Unicorn and the Wasp* — Just like Agatha Christie, you should be able to sift through all the information and find the important things. Are you able to filter information from the webpage to get only the relevant data?
- ii) **EXTERMINATE! EXTERMINATE!** — The CIA factbook has some unstructured data - not all of it is organized properly. This may not be as annoying (or life threatening) as Daleks, but your programs should be able to deal with this correctly and not crash (or get exterminated). Completing this quest will reward you with 30 XP and unlock Achievement: Torchwood. This quest has two tasks: 1) *Partners in Crime or your Companion* — You can get help for parsing through the HTML stuff - you could do it yourself, you could use regular expressions, you could use an external HTML parsing library. Regardless of who your partner in crime is, are you sure that it's working as expected and not accidentally removing or keeping information that you would or wouldn't need, respectively? 2) *Blink* — Your program doesn't need to be afraid of the Angels and can blink, i.e., take longer than a few seconds to run and get all the information. However, this shouldn't be too long, say 1 hour. Does your program run in a reasonable amount of time?

iii) **The Sonic Screwdriver** — This is a useful tool used by the Doctor to make life a little bit easier. Does your code make it easy for you to answer the required questions? Completing this quest will reward you with 40 XP. This quest has three tasks:

1) *Human Nature* — It might be human nature to hard code certain pieces of information in your code. But your code needs to be generic enough to substitute the italicized parts of the questions? Is this possible? 2) *The Sontaran Stratagem* — For some of the questions, you don't need a clever strategy (or algorithm). But for some of the latter questions, you do. Do you have a good code strategy to deal with these? 3) *Amy's Choice* — You have a choice of 2 wild card questions. Did you come up with an interesting question and answer it?

3.3. Assignment 4 and quests

3.4. Student-created HALO Quests

The final course assignment allowed students to design their own projects, making it difficult for us to pre-define HALO quests, since each project was different. Instead, students were offered extra credit in exchange for creating HALO quests for their projects, thus emphasizing the “learning by example” pedagogy. Out of the 124 students who submitted Assignment 5, 77 students (62.1%) attempted the extra credit, and 71 out of these 77 students (92.21%) got a perfect score for the HALO quests that they had created.

We now describe one of the HALO quests that some students created for their own project. This highlights that students understood the basics of software testing, which was the goal with HALO. We include a short description of the project (quoted from student assignment submissions) along with the quests, since students could define their own project.

“**Drawsome Golf:** Drawsome Golf is a multi-player miniature golf simulator where users draw their own holes. After the hole is drawn, users take turns putting the ball towards the hole, avoiding the obstacles in their path. The person who can get into the hole in the lowest amount of strokes is the winner. There are four tasks to complete for the quest for Drawsome Golf: 1) Perfectly Framed (Task): Is the panel for the hole situated on the frame? Is there any discrepancy between where you click and what shows up on the screen? Is the Information Bar causing problems? 2) Win, Lose, or Draw (Task): Are you able to draw lines and water? Are you able to place the hole and the tee box? Can you add multiple lines and multiple ponds? Could you add a new type of line? 3) Like a Rolling Stone (Task): Does the Ball Move where it is supposed to? Do you have a good formula for realistic motion of the ball? 4) When We Collide (Task): Does the Ball handle collisions correctly? Is the behavior correct for when the ball hits a line, a wall, the hole, or a water hazard?”

4. Conclusion

In this paper, we described a competitive-collaborative approach for teaching a CS2 class. The competitive aspects that we used in the class leveraged a code tournament and HALO to teach students good software design and the basics of software testing. The collaborative aspects allowed students to work in teams to create a project of their choice. This resulted in many interesting projects, including some socially relevant and arty projects. Finally, a fun, informal, and collaborative classroom was used to prevent creating stereotypes that CS is boring. The overall student feedback for the class was very positive. Most of the students enjoyed the competitions, the HALO quests, and the team projects and many students decided to major in CS after the class.

Acknowledgment

We would like to thank all the students who participated in the COMS 1007 class. We would also like to thank the Teaching Assistants (Lakshya Bhagat, Amrita Mazumdar, Paul Palen, Laura

Willson, Don Yu) for helping with the class. The authors are members of the Programming Systems Laboratory funded in part by NSF CCF-1161079, NSF CNS-0905246, and NIH U54 CA121852.

References

- [1] J. Bell, S. Sheth, and G. Kaiser. Secret Ninja Testing with HALO Software Engineering. In *Proc. of the 4th Intl. workshop on Social software engineering*, SSE '11, pages 43–47, 2011.
- [2] Blizzard Entertainment. World of Warcraft. <http://us.battle.net/wow/en>.
- [3] Columbia Engineering — The Fu Foundation School of Engineering and Applied Science. Bulletin 2011-2012. <http://bulletin.engineering.columbia.edu/files/seasbulletin/2011Bulletin.pdf>, 2011.
- [4] S. H. Edwards. Rethinking computer science education from a test-first perspective. In *18th annual ACM SIGPLAN conf. on OO programming, systems, languages, and applications*, OOPSLA '03, pages 148–155, 2003.
- [5] S. Elbaum, S. Person, J. Dokulil, and M. Jorde. Bug Hunt: Making Early Software Testing Lessons Engaging and Affordable. In *Proc. of the 29th Intl. Conf. on Software Engineering*, ICSE '07, pages 688–697, 2007.
- [6] U. Jackson, B. Z. Manaris, and R. A. McCauley. Strategies for effective integration of software engineering concepts and techniques into the undergraduate computer science curriculum. In *Proc. of the 28th SIGCSE tech. symp. on Computer science education*, SIGCSE '97, pages 360–364, 1997.