– Independent Work Report Spring, 2017 –

The COS 226 Game: A Gamification Platform for Algorithms and Data Structure Topics

Ragy Morkos
Princeton University, Department of Computer Science, Class of 2018
Adviser: Professor Robert Fish

Abstract

The enrollment for introductory computer science classes has been continuously on the rise throughout universities. This has been causing a problematic strain on computer science departments and faculty, leading to an unavoidable lack of instructor one-on-one presence. Coupled with the difficulty of such introductory classes, especially to novices who have no prior computer science experience, these classes can cause a challenge to a lot of students. Moreover, recent psychological research has shown that college students have a higher tendency to procrastinate due to the proliferation of social media and more hedonic minor activities available at their fingertips on the internet. The problems in introductory computer science classes coupled with the increasing tendency for college students to procrastinate provides a major problem for computer science departments and instructors as well as for the students.

In this paper, we explore the possibility of a gamification environment for learning introductory computer science algorithms and data structures. There has been a plethora of scientific literature on the benefits of learning gamification. Our aim is to take advantage of the techniques in the literature, which are mostly aimed at k-12 education, and adapt them to introductory computer science concepts. Our environment is not aimed at being integrated in the official classroom structure, but rather act as a supplement. Moreover, since we are offering a game based on computer science topics, our intention is to provide a "positive" form of procrastination for students as opposed to social media consumption. Our feedback from students has been largely positive, signifying the useful niche that our gamification environment can fill in computer science college education today.

1. Introduction

Computer science class enrollments in U.S. universities have been seeing a continuous, unprecedented surge. This trend can be seen in universities across the country. For instance, CS50, Harvard University's introductory CS class, has seen its enrollment rise from less than 150 students in 2006-2007 to about 700 students in 2013-2014, an increase of 300% in just seven years [1]. Stanford's enrollment for three of its introductory CS classes rose from 1,000 (2006-2007) to 2,000 students (2013-2014), an increase of 100% [1]. MIT's enrollment for two of its introductory CS classes rose from 500 (2006-2007) to 1,100 students (2013-2014), an increase of 120% [1]. This trend is not just present at elite, private institutions either; University of Michigan's and University of Washington's enrollments for each's main introductory CS class rose 200% and 83%, respectively, between 2006-2007 and 2013-2014 [1]. At Princeton University, where the study for this paper was conducted, the number of computer science degrees conferred rose from 18 degrees in 2006-2007 to 135 in 2016-2017, an astonishing rise of more than 650% in ten years [2, 3]. As of the class of 2017, computer science is the most popular major on campus [3]. Overall at Princeton, approximately 60% of the undergraduate student body would have taken a CS class throughout their studies [4], and approximately 50% would have taken COS 126, Princeton's introductory computer science class [3].

This continuing trend of increasing enrollments in computer science classes and departments in universities throughout the country is forming a supply and demand enigma to the higher education institutions: there simply aren't enough faculty for this seemingly never ending increase in enrollments. One example is Boston College, which witnessed an atrocious reduction in its faculty to student ration from 1:5 in 2007 to 1:34 in the 2016-2017 school year [5]. Introductory CS courses get filled up quickly, making it harder for freshman to explore the field

of CS while any upper level CS courses were restricted only to majors [5]. Yale University suffers from the same problem as their faculty shortage cannot sustain the increasing CS enrollments [6]. Assistant Professor of Computer Science David Kauchak at Middlebury College mentioned how his department is “running at bare bones capacity right now” [7].

This is a problematic trend that, apart from causing stress to universities and CS departments around the country, is also problematic to the students enrolled in CS classes, especially introductory ones. Surprisingly, while enrollment rates are high, retention rates are also high. Numerous research points out how freshmen college students are unprepared for the mathematical and logical rigor that these introductory CS classes need [8, 9]. Furthermore, it has been shown that the availability of positive teaching assistants is crucial to the success of students and the persistence of declared CS majors to stay in the department since students are more likely to struggle at the beginning of such introductory classes [10]. Even more crucial is friendly faculty-student interaction, where the availability of faculty to be able to discuss the major more in-depth as well as provide guidance on assignments, etc. is one of the determining factors for students dropping out of the class or even the CS major [10, 11].

It is also worth noting that this is especially the case for students with no prior experience in programming or CS concepts; students who reported no prior experience before enrolling in such classes were much more prone to drop the introductory class, perform poorly, or switch their major entirely from CS [10, 11]. These students tended to label their CS coursework to be “excessive,” as opposed to their peers who had prior experience in CS before taking the introductory class [11]. The difficulty of introductory CS classes to college students without prior experience, the lack of faculty-student interaction and teaching assistants caused by

shortage can lead to an overall negative experience and frustration for some students, causing them either to have a lower grade, drop the class, or switch to another major altogether [10, 11].

To make matters worse for struggling students in these introductory CS classes, recent research has shown that modern college students in general tend to have extremely high rates of procrastination [12]. In one procrastination study of 700 college students in Germany, it was found that an average college student spends 73 minutes per day or 7 to 8 hours per week browsing Facebook among other forms of social media [12]. Procrastination via social media was so prevalent that only 9% of students reported *not* having used Facebook to procrastinate during the last 6 months [12]. It should be noted that this high rate of procrastination is just in the general college student population. More research points out that procrastination is higher with students who have feelings of academic frustration, overwhelm, stress, and lack of motivation towards the material [13, 14]. Unsurprisingly, procrastination, and especially via social media consumption, is correlated with more negative feelings including anxiety, depression, lower life fulfillment among a plethora of other negative emotions [15, 16].

Multiple educational approaches and improvements to students' learning experiences have been proposed in the scientific literature. One of the biggest of such approaches is learning gamification, in which game elements and design are used in a learning environment or setting. The most famous gamified learning platform is perhaps Duolingo[1], a website and app that aims to teach the basics of languages via game mechanics. Learning gamification, as a pedagogical approach, has been shown to provide very beneficial results to students [17, 18, 19, 20]. Student engagement with and attentiveness to the material was shown to increase using such gamification elements as badges, levels, points, scores, and leaderboards, among others [21].

[1] https://www.duolingo.com/

While a lot of the scientific literature on the topic was concerned with K-12 education, the benefits of learning gamification were shown to reach college students as well [20, 22, 23].

This paper aims to provide the groundwork for addressing the aforementioned problem in computer science college education using gamification of learning. To do so, we have developed a simple website that utilizes gamification mechanics in order to offer a game-like website for COS 226, Princeton University's introductory class to algorithms and data structures. This class was chosen as it is one of the most challenging courses at Princeton with many algorithm running times and data structures that students need to keep track of. This made the course especially suitable for the game model we developed of multiple choice questions based on COS 226 material. Our aim is to provide students who would otherwise procrastinate on social media, as mentioned previously, with a more positive form of procrastination where they are passively learning the material through playing our game, especially since our website was designed as a nonofficial tool for students to use.

While there are many problems that can lead to less than desirable student performance in introductory computer science classes, including faculty shortage, lack of student incoming experience, social media procrastination tendencies, etc., and while these problems are complicated and intertwined, our proposed game website can offer a tool in the toolkit of each student in order to help them improve their performance. Our proposed game website was therefore designed to explore the future and usefulness of gamification pedagogical techniques for students in introductory CS classes. Our feedback from students has been largely positive for such a simple website, boding well for the promise of gamification in learning introductory CS topics.

Our paper is organized as follows. In Section 2, we discuss related work and background, most notably previous learning gamification trials or studies specific to CS or engineering education. We will take a closer look at examples of what a gamified learning environment actually entails and how effective these methods are in student satisfaction and learning progress. In Section 3, we discuss our approach to developing our game website, including but not limited to features, score calculations, aesthetic elements, question format, among others. In Section 4, we provide in detail our implementation including details of our used backend, frontend, and database technologies and languages. In Section 5, we provide our evaluation results for the small sample of students that tried our gamification system. In Section 6, we discuss future work and how our gamified learning environment can be further enhanced, tweaked, as well as expanded to a larger user base. Finally, we conclude in Section 7.

2. Related Work

The research made on gamification of learning in engineering and computer science education is plentiful and consistently growing. Because of this, we have only picked the most interesting research on gamification of engineering and CS education to discuss in this section. Nonetheless, a careful reading of all the related work on the topic is encouraged as each study offered insight into how gamification can be helpful to computer science students [28, 29, 30, 31, 32, 33, 34, 35, 36].

2.1 Instituto Superior Técnico: Engaging Engineering Students with Gamification

An interesting study done at Instituto Superior Técnico in Portugal compared student performance in a CS class on different metrics in two years, where one year the class utilized a

banal course website with lecture slides, assignments, etc. and the following year the course website adopted gamification elements in its interface [24]. Gamification elements included experience points, levels, leaderboards, challenges, and badges. Comparisons between the class offerings in the two years were made based on student attendance, participation, usage of reference materials, as well as student feedback surveys.

The results for the study were largely positive, pointing to the usefulness of even simple gamification elements on the online learning platform for the course. For lecture slide downloads, the average download per lecture rose approximately 50% from the previous year [24]. Student participation on the course website in terms of posts, replies, student interaction, etc. rose an astounding 845% despite the lower enrollment of students in the second year [24]. Furthermore, faculty participation increased 325% from the previous year, an increase that can be attributed to how faculty members would make more replies since students have posted more [24]. Attendance in the second run of the CS class also experienced higher attendance rates, with an increase of 11% from the prior year [24]. Quantitative and qualitative feedback data from students was largely positive, with the most noteworthy feedback being that students complained less about the workload, despite the workload being exactly the same as the previous year's workload. This suggests that students might have found the gamified environment of their course more enjoyable that they felt the rigor of the material much less [24].

2.2 Aalto University: Effect of Achievement Badges on Students' Behavior

Another interesting study done at Aalto University in Finland compared two groups of students in an algorithms and data structures class [25]. One group of students in the class were exposed to different achievement badges in the online learning environment, while the other

group didn't have achievement badges in their learning environment. The learning environment in both groups was used for the small graded exercises that are part of the course grades [25]. Performance of students in the group with the achievement badges was higher than in the group without, with 74% of students in the group incorporating the achievement badges into their learning environment mentioning that they found the badges motivating. Moreover, students in the experimental group tended to submit the exercises later than their peers in the other group due to wanting to answer more correctly and obtain the badge [25]. Finally, 79% of students in the experimental group mentioned that they think the badges feature on the learning environment should persist in the course [25]. These are numbers are quite telling of the positive impact even a small feature such as badges can have on the learning experience of students.

2.3 University of Colorado: A Dissertation on Gamification in Introductory CS

Perhaps the most extensive study on the topic was a dissertation done at the University of Colorado [27]. The study comprised multiple ages and demographics and the gamification of the material was done in multiple ways. In one way, the student progress was gamified to a great extent, including "missions", "quests", "badges", and even "boss fights." 61% of students who used this gamified environment felt that that it made the course more interesting [27]. In another less gamified environment, with simple gamification features such as a leaderboard, animation, as well as puzzle pieces corresponding to programming blocks, 50% of the students explicitly mentioned that the gamification style of the environment was a positive part of their experience. Overall, the extensive research carried out by the dissertation concluded that complex and even simple gamification features produced a positive experience for students and helped them learn more [27].

3. Approach and Design

3.1 Website Design Decision

Our game environment for COS 226 was developed on a simple website[2] on Princeton's CS department's servers. The reasons for developing our game environment on a website rather than an app are twofold. First, we used Princeton's CAS (Central Authentication System) as the login method for our website. This would make trying out the website more effortless for students since all Princeton students already use their login credentials for a host of other Princeton services. Because of this, we refrained from making this as a mobile app since there are complications in using Princeton's CAS on mobile systems. Moreover, having the environment as a downloadable app would require more effort on the part of students, which might decrease students' willingness to try our environment out.

Second, from the psychological research on procrastination [12], it seems that while students do use mobile phones in their procrastination, the vast majority actually use their desktops or laptops more, since opening a Facebook or YouTube tab on their browser while writing a paper or studying for an exam is much easier than explicitly using their phone for the same purpose. Since our game website was designed as a form of positive procrastination, we opted for the website design decision so that students would have the easiest access to our environment.

3.2 Levels, Question Format and Timing

Our question format for the website was based on a simple multiple choice question format based on facts from the COS 226 textbook. Questions did not require any calculations or

[2] http://rmorkos-226.cs.princeton.edu/

explicit computations, but rather required that the student would know how a simple algorithm or data structure works or its running time or space usage. We opted for this format since this requires the least amount of effort on the part of students and so it would incite more participation and usage of the website.

There are multiple “levels” on the website, each one corresponding to a certain lecture topic on the COS 226 curriculum. Each level is comprised of three questions randomly generated from a bank of questions on that topic. The user has 45 seconds to answer the three questions correctly, and if the timer runs out, the answers the user marked are automatically submitted. The “score” on each level that the user obtains is a combination of how quickly they answer the questions as well as how correctly they do. Furthermore, multiple checks are placed such that users cannot increase their score by simply selecting random answers. The timing mechanism was implemented as a gamification technique for eliciting user interest and making the playing experience more intense.

3.3 Leaderboard and Student Scores

The homepage for the game website includes the student’s total score on all the levels as well as a breakdown of their score on each level. This is put side to side with the current leaderboard scores, where the highest total score achieved is shown on the website as well as the highest score achieved for each level. Putting both scores (student’s and leaderboard’s) side by side was done on purpose in order for the student to gauge their performance with that of their peers. Moreover, from the feedback obtained in the different gamification of learning studies [20, 21, 24], students found ratings, scores and leaderboards to be effective features to making them more competitive and encourage them to spend more time practicing and playing in the

environment to improve their scores and position in the leaderboard. To that end, we decided to put this information on the homepage for each user.

3.4 Game Mechanics, Feedback, and Options

Upon entering the homepage, the student will find a simple description of the game and how to proceed. On the bottom right corner would be their current scores, and on the bottom left would be the leaderboard scores. The student would have a button labelled “Play!” that will take them through the first level of the game. Upon finishing the level, they will be automatically directed to the “Results” page, where they will find feedback for every question they got wrong alongside with their answer and the correct answer. At the bottom of the page would be the option button to replay the level “Play Again?” or the option button to “Go to Next Level!” At the last level of the game, students would have the option to “Go Back to Welcome Page!” instead of the one of going to the next level.

4. Implementation Details

4.1 Website Technology

Our website was extremely simple to develop, using PHP for backend processing, HTML, CSS, and JavaScript for website aesthetics as well as client-side functionality. All information, including question bank, answers, usernames, user scores, and leaderboard scores, were stored and processed using SQLite. Our website could have been developed using a number of other languages or technologies, so our design decision ultimately came down to using the simplest ones along with the ones that would be easiest to maintain by future developers of the website.

4.2 Homepage Features

This subsection will discuss in detail our homepage design and features. Figure 1 shows a screenshot of the homepage.

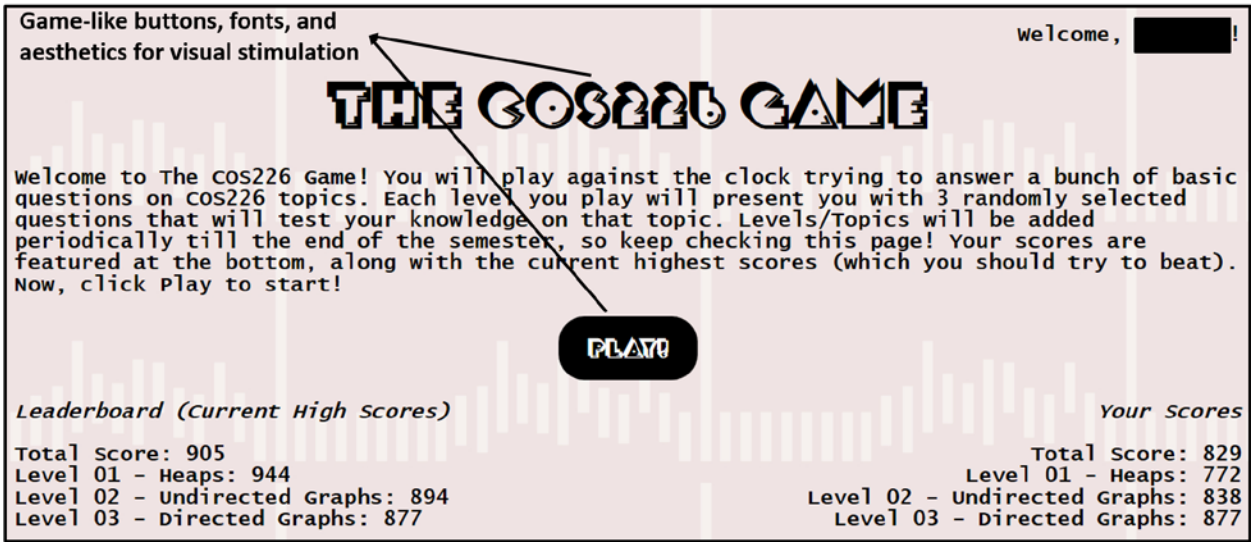

Figure 1: Implementation of Homepage

At the top right-hand corner of the homepage is a greeting of the user to confirm his successful login; the username is obtained from Princeton's CAS. The background throughout the website is adopted from the cover of the COS 226 textbook[3] to give an aesthetic style to that of COS 226 material. For the major titles of the website as well as the buttons, we utilized a Pac-Man freely available font[4] in order to give the website gamified aesthetics. The homepage's introduction describes the game and how it is scored. At the bottom right and left corners are the student's current scores and leaderboard scores, respectively.

[3] http://www.cs.princeton.edu/courses/archive/spring17/cos226/images/algorithms-cover.png
[4] http://www.fontspace.com/font-a-licious/pacfont-good

4.3 Implementation of Levels and Questions

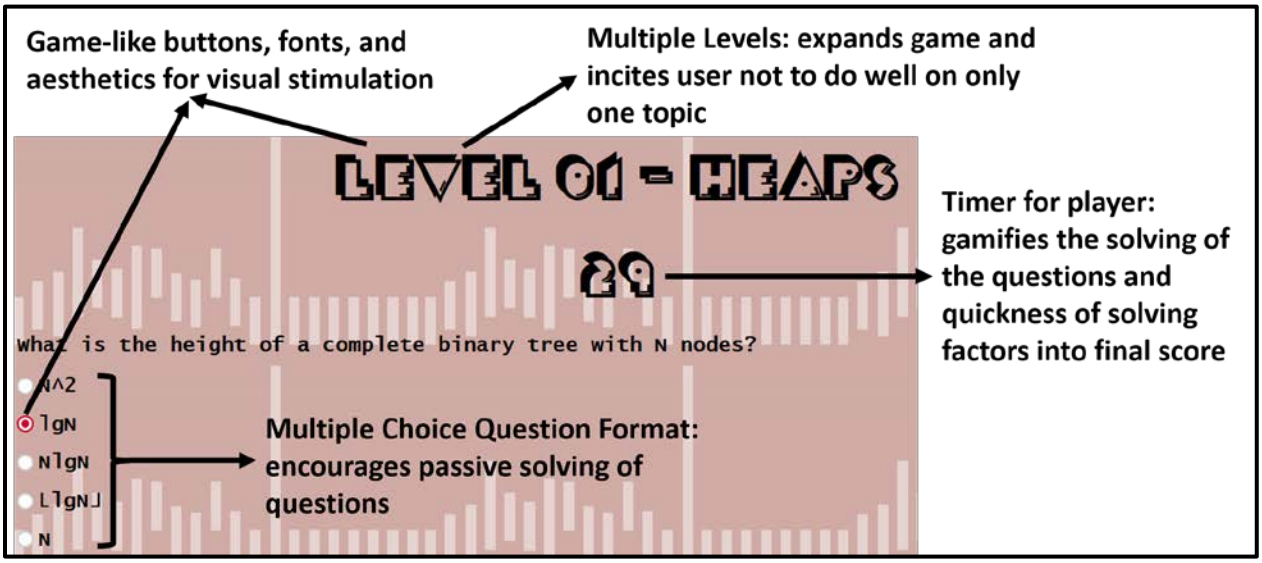

Figure 2: Example of Level Implementation

Figure 2 shows a screenshot of one of the game's levels. As with the rest of the website, the same background and font for titles is used for consistency. We can see in detail a look at the question format of the website. As shown in the screenshot, the multiple choice question format encourage passive solving by the students since no quantitative work is required on their part. This also helps make the questions more realistically solvable in the allotted time. Furthermore, the questions are relatively easy, requiring only a basic familiarity of the data structures and algorithms the question asks instead of in-depth knowledge on how it works.

4.3 Implementation of Student Feedback

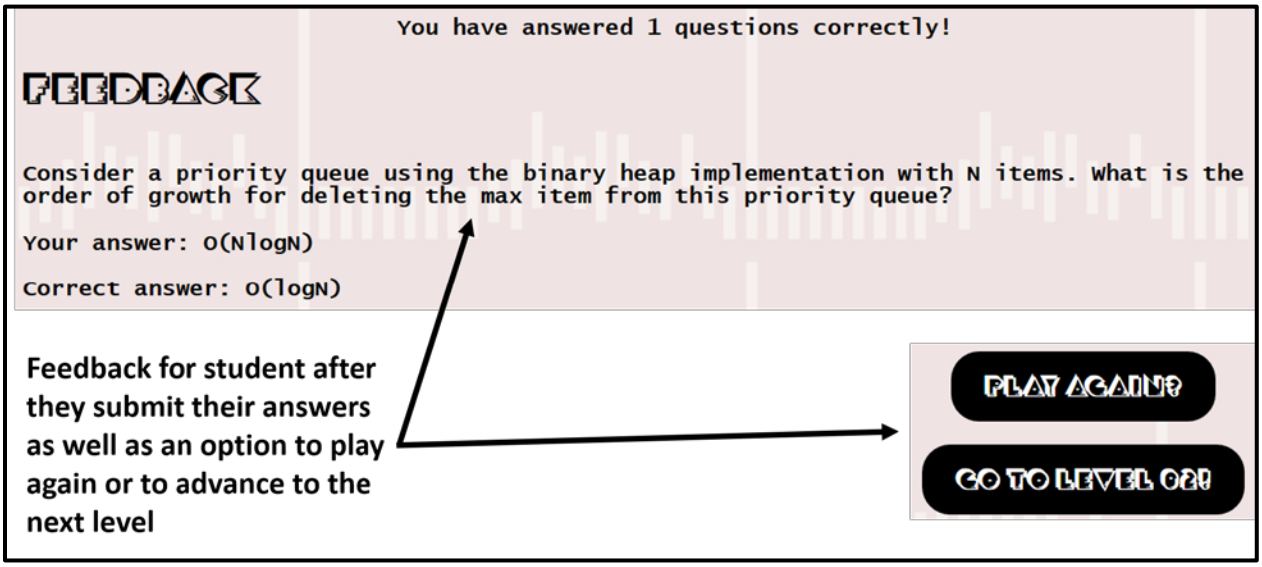

Figure 3: Example of Feedback Page

Figure 3 shows a closer look at the feedback page for one of the game's levels. Students can see how many questions they got wrong, as well as what those questions were alongside with the student's wrong answer and the correct answer for the question. After reviewing their performance, students will have the option to either replay that level, in which they will be directed to that level's page with newly selected questions, or they can proceed to the following level.

5. Evaluation Results

For our evaluation, we asked student volunteers who were taking COS 226 to try out the website and provide their feedback via a simple Google form. At the end of the survey period, 16 students had given their feedback about the website via the Google form. A screenshot of the form could be found in Figure 4.

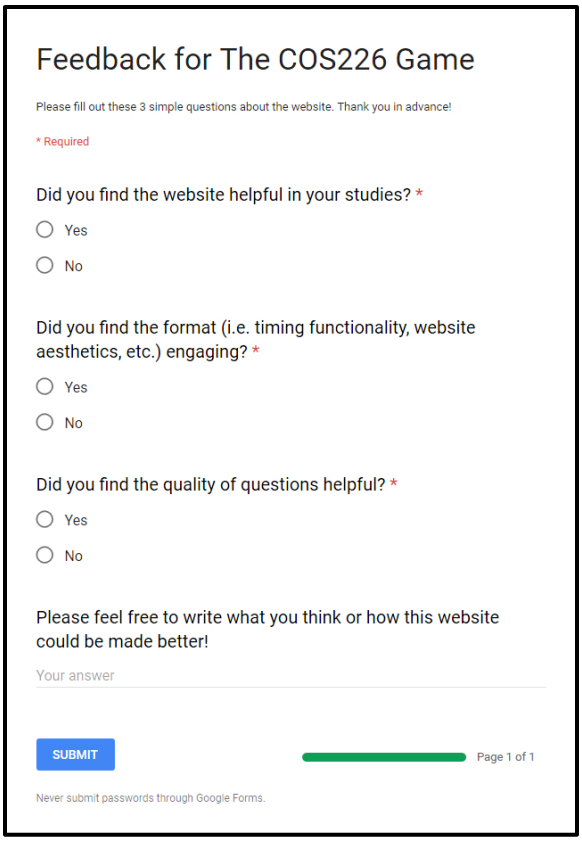

Figure 4: Screenshot of Student Feedback Form

For the first question, approximately 69% of students reported the website being helpful in their studies. For the second question, approximately 69% of students have found the website aesthetics to be engaging. Finally, 69% of students have found the quality of the questions to be helpful.

While most students opted not to provide written comments in their feedback, there were some interesting suggestions that a few students made. One student found the questions helpful, but commented on how the given 45 seconds for each question were not enough to solve the questions. Another student requested an explanation feature where the feedback given after each level would not only show the wrong and right answers, but also offer explanation for the correct

answer. Finally, a student suggested different “modes” where questions of different types would be offered instead of the simple factual questions currently present at each level.

6. Future Work

Gamification of learning is still a growing field, especially for college level material, and especially more for computer science college material. Through our study in this paper we have discovered that there exists potential work that could further improve computer science education using gamification techniques. As shown in our project, gamification need not be implemented in the official classroom to provide benefits to students. Even when used as an unofficial classroom source of study, it can have beneficial results for students.

The COS226 Game has a large room for improvement. As a start, the question bank definitely has room for growth and increase in the questions offered at each level. Furthermore, the question format could definitely use variety based on the student feedback we obtained. This task would require the cooperation of multiple other student developers as well as possible faculty involvement with the question production process. Another related major feature that could be added to the website are explanations offered for questions that the student answers incorrectly. This would also require more pedagogical expertise with possible involvement of the course’s faculty.

One more improvement on the website could be made in its aesthetics. Currently, the website design while minimalistic and fairly aesthetic, could still use more editing. This task would require the incorporation of more advanced web development technologies and libraries. Similarly, game audio, badges, and more gamification features could definitely be added to the website in order to enhance it even more.

Another major feature that can be added to the website is a “multiplayer” mode, where two students who are currently online can be directed to a page to answer COS 226 questions. Each student will have a unique queue of questions to answer, and each correct answer of a question gets that question removed from the queue. If a question gets answered incorrectly, the student gets another question added to their queue. The student who finishes their queue first wins. This multiplayer feature is well-known in many simple online games [26], boasting incredibly high user addiction and participation. While can this have negative effects in another setting, having students highly engaged in playing against each other on an educational website like The COS226 Game is a positive use of procrastination that students can make.

7. Conclusion

In this paper, we have proposed a gamification environment for COS 226 (Algorithms and Data Structures). As stated in the introduction, computer science education around the country is experiencing many problems, including rising student enrollments, faculty shortage, and high drop-out rates either from introductory classes or even declared majors. Because gamification of learning was shown in our related work to offer a high potential for positive student participation and increased student satisfaction, we have opted to develop our gamification environment for COS 226 material as an unofficial source for class material and practice. Through our gamified implementation of the website and its content, we tried as much as possible to adopt a gamified approach to learning algorithms and data structures, an approach that the current scientific literature shows could offer a multitude of unearthed benefits that computer science students in colleges could greatly make use of in the face of difficulties currently faced by CS departments around the country.

8. Honor Code

This paper represents my own work and is in accordance with Princeton University's Honor Code.

Ragy Morkos

9. References

[1] Lazowska, Ed, Eric Roberts, and Jim Kurose. "Tsunami or Sea Change? Responding to the Explosion of Student Interest in Computer Science." NCWIT 10th Anniversary Summit 1 (2014).

[2] "Princeton University Degrees Conferred, 1970-Present." Princeton University Office of the Registrar. Web. 28 Apr. 2017. https://registrar.princeton.edu/university_enrollment_sta/degconf.pdf

[3] Li, Jessica. "COS now most popular major." The Daily Princetonian. The Daily Princetonian Publishing Company, 23 Sep. 2015. Web. 28 Apr. 2017. http://www.dailyprincetonian.com/article/2015/09/cos-now-most-popular-major

[4] Wolff, Josephine. "On the Rise: Computer science becomes top major, expanding faculty as it transforms fields." Princeton Alumni Weekly. The Trustees of Princeton University, 20 Apr. 2016. Web. 28 Apr. 2017. https://paw.princeton.edu/article/rise-computer-science-becomes-top-major-expanding-faculty-it-transforms-fields

[5] Confalone, Leo. "Understaffed, Computer Science Seeks Faculty to Meet Demand." The Heights. The Heights, Inc., 24 Apr. 2017. Web. 28 Apr. 2017. http://bcheights.com/2017/04/24/understaffed-computer-science-seeks-faculty-meet-demand/

[6] Rogers, Stephanie. "CS department short on faculty." Yale Daily News. The Yale Daily News Publishing Company, 21 Oct. 2014. Web. 28 Apr. 2017. http://yaledailynews.com/blog/2014/10/21/cs-department-struggles-for-faculty/

[7] Henriques, Will. "Computer Science Faculty Seek Expansion." The Middlebury Campus. 03 Oct. 2012. Web. 28 Apr. 2017.

[8] Hewner, Michael. "Undergraduate conceptions of the field of computer science." Proceedings of the ninth annual international ACM conference on International computing education research. ACM, 2013.

[9] Beaubouef, Theresa, and John Mason. "Why the high attrition rate for computer science students: some thoughts and observations." ACM SIGCSE Bulletin 37.2 (2005): 103-106.

[10] Barker, Lecia J., Charlie McDowell, and Kimberly Kalahar. "Exploring factors that influence computer science introductory course students to persist in the major." ACM SIGCSE Bulletin. Vol. 41. No. 1. ACM, 2009.

[11] Biggers, Maureen, Anne Brauer, and Tuba Yilmaz. "Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers." ACM SIGCSE Bulletin. Vol. 40. No. 1. ACM, 2008.

[12] Meier, A., Reinecke, L., & C.E. Meltzer (2016). “Facebocrastination”? Predictors of using Facebook for procrastination and its effects on students’ well-being. Computers in Human Behavior, 64, 65-76.
[13] Grunschel, Carola, Justine Patrzek, and Stefan Fries. "Exploring reasons and consequences of academic procrastination: an interview study." European Journal of Psychology of Education 28.3 (2013): 841-861.
[14] Brownlow, Sheila, and Renee D. Reasinger. "Putting off until tomorrow what is better done today: Academic procrastination as a function of motivation toward college work." Journal of Social Behavior and Personality 15.5 (2000): 15.
[15] Tripathi, Kamana, and Krishna Sharma. "Causes of academic stress among college students and its managements." Indian Journal of Health and Wellbeing 4.5 (2013): 1161.
[16] Jafri, H. "A Study on Facebook Addiction and its Relationship with Emotional Experience and Coping Strategies." The International Journal of Indian Psychology 2.4 (2015): 7-14.
[17] Paiva, José Carlos, José Paulo Leal, and Ricardo Queirós. "Gamification of learning activities with the Odin service." Computer Science and Information Systems 00 (2016): 25-25.
[18] Gros, Begoña. "The impact of digital games in education." First Monday 8.7 (2003): 6-26.
[19] Cheong, Christopher, Justin Filippou, and France Cheong. "Towards the gamification of learning: Investigating student perceptions of game elements." Journal of Information Systems Education 25.3 (2014): 233.
[20] Gudoniene, Daina, et al. "Technological Aspects of the Gamification Model for e-Learning Participant's Engagement." Baltic Journal of Modern Computing 4.4 (2016): 1008.
[21] Brull, Stacey, and Susan Finlayson. "Importance of Gamification in Increasing Learning." The Journal of Continuing Education in Nursing 47.8 (2016): 372-375.
[22] Dicheva, Darina, et al. "Gamification in education: A systematic mapping study." Educational Technology & Society 18.3 (2015): 75-88.
[23] Barata, Gabriel, et al. "Improving participation and learning with gamification." Proceedings of the First International Conference on gameful design, research, and applications. ACM, 2013.
[24] Barata, Gabriel, et al. "Engaging engineering students with gamification." Games and Virtual Worlds for Serious Applications (VS-GAMES), 2013 5th International Conference on. IEEE, 2013.
[25] Hakulinen, Lasse, Tapio Auvinen, and Ari Korhonen. "Empirical study on the effect of achievement badges in TRAKLA2 online learning environment." Learning and Teaching in Computing and Engineering (LaTiCE), 2013. IEEE, 2013.
[26] Ducheneaut, Nicolas, et al. "Alone together?: exploring the social dynamics of massively multiplayer online games." Proceedings of the SIGCHI conference on Human Factors in computing systems. ACM, 2006.
[27] Behnke, Kara Alexandra. Gamification in Introductory Computer Science. Diss. University of Colorado Boulder, 2015.
[28] Swacha, Jakub, and Paweł Baszuro. "Gamification-based e-learning platform for computer programming education." X World Conference on Computers in Education. 2013.

[29] Dubois, Daniel J., and Giordano Tamburrelli. "Understanding gamification mechanisms for software development." Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering. ACM, 2013.
[30] Tillmann, Nikolai, et al. "Code Hunt: Gamifying teaching and learning of computer science at scale." Proceedings of the first ACM conference on Learning@ scale conference. ACM, 2014.
[31] Browne, Kevin, and Christopher Anand. "Gamification and serious game approaches for introductory computer science tablet software." Proceedings of the First International Conference on Gameful Design, Research, and Applications. ACM, 2013.
[32] Morrison, Briana B., and Betsy DiSalvo. "Khan academy gamifies computer science." Proceedings of the 45th ACM technical symposium on Computer science education. ACM, 2014.
[33] Li, Cen, et al. "Engaging computer science students through gamification in an online social network based collaborative learning environment." International Journal of Information and Education Technology 3.1 (2013): 72.
[34] Schäfer, Andreas, et al. "From boring to scoring–a collaborative serious game for learning and practicing mathematical logic for computer science education." Computer Science Education 23.2 (2013): 87-111.
[35] Ibáñez, María-Blanca, Angela Di-Serio, and Carlos Delgado-Kloos. "Gamification for engaging computer science students in learning activities: A case study." IEEE Transactions on Learning Technologies 7.3 (2014): 291-301.
[36] Sheth, Swapneel Kalpesh, Jonathan Schaffer Bell, and Gail E. Kaiser. "Increasing student engagement in software engineering with gamification." Columbia University Computer Science Technical Reports (2012).