Gamified website for the Computer Science e-learning environment in Hong Kong

CAES9542 Technical English for CS
Interim Report
Lam Chun Ho 3035685184
fyp22003

January 21, 2023
Abstract

There are many online resources and tutorials for students to access. The assessments and exams are sometimes similar to previous years or questions from the internet. Students may reference the solution from the internet simply and lack interactions with others.

There are a few discussions and forums about Computer Science in Hong Kong. In Moodle forum, Teachers are mostly the respondent to the questions. Therefore, the learning environment in Computer Science is unmotivating, and students may lack the knowledge of Computer Science and perform weaker than other employees in the future.

This project designs a gamified website that applies game elements to non-game contexts. Students can get points and badges by joining contests, finishing courses, solving coding challenges, and discussing in forums. Students are mainly ranked and leveled into three categories, Evil, Code Reviewer, and Adventurer. With gamification, students can be motivated to learn Computer Science and prepare for future careers. Compared to LeetCode, it is an educational website to learn programming instead of a tool for students to practice for interviews.

The project team uses Judge0 and Strapi to provide the services of the website with the Next.js framework. Currently, the project team designed the pages in Figma, where the features and designs reference from LeetCode (discussion, contests, and problems), Reddit (forum), and Moodle (course list). The project team implemented the Challenges and Forum pages based on the designs. The future steps of this project will be implementing the badge, leveling, and points system. The project team will develop the leaderboard, coding contest, user profile, and course pages after that.

Given the time limit, this project does not evaluate the effectiveness of the gamified website. Further actions will be examining the effectiveness of the gamified website. This project also does not build some services from scratch. For example, the project team uses open source Content Management System and Remote Code Execution System, Strapi and Judge0.
Acknowledgments

I would like to thank my supervisor Dr. Choi, Yi King for supervising my project.
# Contents

Abstract ................................................................. i  
Acknowledgements .................................................. ii  
List of Tables ......................................................... v  
List of Figures ......................................................... vi  
List of Abbreviations ............................................... vii  

1 Introduction ......................................................... 1  
1.1 Background ....................................................... 1  
1.2 Problem Statement ............................................... 2  
1.3 Objectives ........................................................ 2  
1.4 Project Scope ..................................................... 3  
1.5 Outline ............................................................ 3  

2 Literature Review .................................................. 4  
2.1 E-learning and Gamification ..................................... 4  
2.2 Design Principles for Gamification ............................. 5  
2.3 Gamified Implementations ...................................... 5  
2.4 Gamification Experiments ...................................... 5  

3 Methodology ......................................................... 7  
3.1 Project Structure .................................................. 7  
3.2 Compared to Modifying Moodle ................................. 7  
3.3 Game Elements ................................................... 8  
3.4 Non-game Contexts .............................................. 10
List of Tables

3.1 Brief comparison to modifying Moodle ....................... 8
3.2 Brief comparison between this website and LeetCode and HackerRank 12
List of Figures

3.1 Project Structure .................................................. 8
3.2 Examples of feedback: Immediate(left); Progress(middle); Statistic(right) 9
3.3 Example Contents in the User Profile, captured from LeetCode ........ 9
3.4 Use cases of the website ........................................... 10
3.5 Example of Code review section .................................. 10
3.6 Flow of the coding challenge .................................... 11

4.1 Lifeline diagram of coding challenge ............................. 14
4.2 The state, parameter and functions in “usePlayer” ............... 15
4.3 The state, parameter and functions in “useMonster” ............ 15
4.4 Flow of worker ..................................................... 16

C.1 Pages of the website ................................................. 30
List of Abbreviations

CMS  Content Management System
CS   Computer Science
HK   Hong Kong
Introduction

For CS students, there are many challenging school assessments. Students can find many online resources without effort. For example, it is easy for students to find tutorials on StackOverflow and YouTube. When it comes to learning, however, these resources are two-sided blades for them. Students may lack the motivation to learn and understand the materials. The following discusses the relationships between the background and problem statement, objectives, scope, and outline.

1.1 Background

There are many overseas forums for discussing different knowledge of CS. For example, students can find solutions for coding in StackOverflow, and mathematics problems in Mathematics Stack Exchange. In China, many forums are in mixed languages, English and their native language, which helps different students to learn. Statistics in StackOverflow 2021 developer server have shown that HK only has 0.24% in the Geography category, which is small compared to China (1.27%) and France (3.25%) [7]. These overseas forums even have active discussions about job seeking, ranging from start-ups to famous companies.

HK students have strong English skills. They may visit course-based forums in Moodle, local large-scale forums like HKGolden and LIHKG, and overseas forums. However, there are no large-scale discussing forums specifically about CS. In contrast to overseas, there are few discussions about CS knowledge in course-based and local, compared to overseas. Students may ask questions in Moodle forums when working on
the assessments and revising exam contents. However, most respondents are teachers in the course-based forum.

Furthermore, the course materials and scope are sometimes similar to previous years because tutors may reference the internet or previous assessments when designing the assignments. Students can seek helpful resources from seniors or the internet to finish them effortlessly. They may effortlessly copy a workable solution to the questions from these resources. As a result, students may not understand the materials clearly and lack interactions with others.

Nowadays, e-learning is a future trend. Many studies evaluate the effectiveness and attempt the possible implementations between e-learning and gamification. Based on the review in chapter 2, gamification can potentially motivate the learning environment of Computer Science and academic participation of students, and may replace the traditional face-to-face lecture.

1.2 Problem Statement

The above shows that the learning environment in CS is unmotivating and inactive. Students may lack the motivation to understand the materials and acquire the answers from the websites without effort because of the powerful websites. They may lack participation in learning algorithms and discussion in such an environment. As a result, students may be weak in future programming works and interviews after graduation. They may need more time to understand the knowledge after graduation and be less competitive than others in famous companies like Google. It is because they do not clearly understand the materials. Thus, this project aims to design a gamified website for the HK CS e-learning environment with the following objectives.

1.3 Objectives

Overall, this project will create a gamified website for university education in programming. The website will transform the non-game features into game elements. Students can, therefore, gain motivation to learn CS through gamification. Students can be leveled and ranked in several categories. They can obtain points and badges by
joining contests, completing coding challenges, finishing courses, and discussing in the forum. The website provides feedback to students, and students are leveled and ranked by their experiences. Chapter 3 will discuss the details. By discussion, thinking, and contributing, students can learn and reinforce their knowledge and prepare for future interviews and careers in their e-learning environment.

Furthermore, this project can be a prototype or example for the future. The approaches and structure can be a direction for the future development of other gamified websites. The way of adding the game elements to non-game contexts can be a package or library to reduce the development process and logic for this project or other websites.

1.4 Project Scope

Given the time limit, the project focuses on Programming. It is because programming is essential in future career and academic learning. Along with this, this project designs a gamified website based on the findings in other research, where it focuses on learning to code and motivating the learning environment. The website does not develop for a single course but for all CS students to use it. This project does not include research on AI or ML but may implement functions with AI or ML. For example, if the project team gives experience to students by using the quality predictor of the reply to a post. Besides, this project does not collect real-life data and research the effectiveness of the design. Moreover, the website may not contain too attractive designs on layout and all functionalities and develop some functions (for example, the chatting system [2]).

1.5 Outline

The structure of this report is as follows: chapter 2, the literature review, describes the idea behind the product and explains the definitions of terms, designs, implementations, and experiences in the research. Chapter 3 is the methodology, which explains the approaches used in the project, including different tools and frameworks to develop the features for the website. In the end, chapter 4 discusses the progress result of the project.
Chapter 2

Literature Review

This chapter describes the idea behind this product with the studies on gamification in e-learning. The following discusses a review of the relationship between e-learning and gamification. Along with this, there are some reviews on the designs, implementations, and experiments in gamification.

2.1 E-learning and Gamification

E-learning is a method for students to learn everywhere and every time within the time limit, which gain higher control and freedom over their learning progress [3]. According to Amriani et al. [1], e-learning should be a tool to encourage participation in the traditional class environment and for students to learn actively.

Gamification is an idea that applies game elements to non-game contexts [4],[6]. The existing examples are Nike+ Run Club, eBay, and Todoist, which used gamification to produce motivations and connections between customers and the company [4]. There are some implementations of gamification in education; a famous language learning app DuoLingo also implemented gamification. Studies [1]-[3],[5]-[6] also add gamification designs and strategies to the e-learning environment to examine the effects on motivating the learning environment, their performances, participation, and more. Students will gain more motivation and engagement with the given missions by combining gamification and e-learning [1].
2.2 Design Principles for Gamification

Game elements are taking an essential role in gamification. There are six common game elements: feedback, goals, badges, points, a leaderboard, and a level system [1]-[4], [5]-[6, Sec. 1]. These game elements follow the three aspects of gamification design [4] (see Appendix A). Moreover, the application should also integrate with classes and courses in education. For example, see [1]-[3],[5]-[6, Ch. 4, pp. 53-78]. Gamification implementations should also consider the self-determination theory [1],[5] (see Appendix B) because of its emphasis on their goals.

2.3 Gamified Implementations

Some researchers implemented gamification in e-learning in Moodle. For example [1],[3],[5]-[6, Ch. 11, pp. 238-260] modified Moodle and applied different game elements to it. There was also an implementation called PeerSpace that provides a collaborative learning environment that includes functions with online social networks [2].

2.4 Gamification Experiments

According to Poondej et al. [3], adding gamification strategies should be considered to motivate and engage students in learning. Most students in the gamification group had positive feedback on gamified education compared to the traditional education group [1]-[3],[5]. The gamification group had higher motivations than the control group [2],[3]. However, Amriani et al. [1] suggested that gamification does not imply evident effects on students’ participation but on their performance. The class materials had higher downloads for the experimental group, which had an average of 89 downloads per week, and more than 65 downloads per week for the control group [5]. The experimental group in [6, Ch. 7, pp. 131-151] had a obviously better academic performance than the control group, and the two groups had a similar score on motivational beliefs. Experiments [1]-[3],[5],[6, Sec. 2] showed that gamified environments may gain positive effects, including motivation for learning and participating, and may also improve academic preferences. That shows the positive
influence of gamification in e-learning, including better academic results and higher intention for learning. Therefore, this project will apply the game elements to the website. The project will implement common game elements to the courses, forum, coding contests, and coding challenges.
Methodology

This chapter describes the methodology, including the non-game contexts, game elements, comparison to existing websites, and comparison to modifying Moodle. Overall, this project uses Next.js as the framework of front-end development. The project team will also use Strapi as the CMS and Judge0 as the Remote Code Executing System to reduce the development complexity and time.

3.1 Project Structure

Figure 3.1 shows the structure of the project. The project team uses Strapi and Judge0 as CMS and Remote Code Execution System for the backend services. They communicate with the presentation layer Next.js with middleware: Redux, and NextAuth.js, which provides an easy way for data fetching and caching tools and an authentication tool. Compared to modifying Moodle, the project structure has higher customizability and scalability. Compared to Nuxt.js and Nest.js, Next.js has a larger community and flexibility for development.

3.2 Compared to Modifying Moodle

As mentioned in Chapter 2, there are some experiments that chose to modify Moodle. However, the project team chose to creating a gamified instead. Table 3.1 shows the comparison between creating this website and modifying Moodle. With the project structure, the project website mainly uses TypeScript and not using Javascript
and PHP. This provides type safety for developers. With the use of CMS, this website is dynamic for different databases, SQL or NoSQL, while Moodle only uses SQL. This website is more than a course-based website. It aims to be a community for university CS students.

<table>
<thead>
<tr>
<th></th>
<th>Creating This Website</th>
<th>Modifying Moodle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>TypeScript</td>
<td>PHP &amp; JavaScript</td>
</tr>
<tr>
<td>Database</td>
<td>SQL or NoSQL</td>
<td>SQL</td>
</tr>
<tr>
<td>Use as</td>
<td>Community</td>
<td>Course-based</td>
</tr>
</tbody>
</table>

Table 3.1: Brief comparison to modifying Moodle

### 3.3 Game Elements

Levels, points, and badges can show the statistic, experience, progress, and achievements of students. The leveling of students will show their experiences in different categories, including programming language, code reviewer, and more; The website can distribute the points to students, including progress points on courses, skills points on programming languages, and experience points on levels. Students can get points by their actions in the website; The badge system can create and distribute the badge to students, which can show the achievement and progress of students. For example, when students finished a course or great contributions to the community, the system will award a badge to them.
The feedback is the visual element of this project. As shown in figure 3.2, there are three types of feedback: intermediate, progress, and statistics. Intermediate feedback shows simple feedback like earning some experience points, “You did it” and “Wrong answer.”; Progress feedback shows the progress of the course, the remaining experience for the next level, and more; Statistics feedback is a visual way to show the students’ performance. For example, students can find their rank in coding contests.

The leaderboard page is the second visual element of this project. Students are mainly leveled and ranked into three categories: Evil, Code Reviewer, and Adventurer. Evil is related to posting coding challenges; Code Reviewer is related to reviewing other students’ code; Adventurer is related to solving coding challenges.

Profile Page

Figure 3.3 shows examples of the profile page. On the profile page of students, students can view their badges, skill points in different programming languages, and their rank in coding challenges.
3.4 Non-game Contexts

Figure 3.4 shows the use cases of the website. There are four contexts on the website. The following will discuss in detail.

![Use cases of the website](image)

**Figure 3.4: Use cases of the website**

3.4.1 Forum

In general, students can discuss anything in the forum. This project also adds a new feature: code review. Students can reply to a post with a code review, which makes a suggestion to part of the post/source code (for example figure 3.5). As a result, students can learn from others and understand their performance. The website only allows students who solved the same coding challenges to review the submissions from each other.

![Code review section](image)

**Figure 3.5: Example of Code review section**

Students can get points by creating a new post or replying to a post. However, it is not fair to give the same points to students who reply very well or very badly.
Therefore, the project team may implement an AI to give a prediction on whether the reply is a good reply or not.

### 3.4.2 Coding Challenges

Figure 3.6 shows the flow of the coding challenge. There are two stages in a coding challenge, similar to having an interview. Firstly, the student should answer how they solve it in their own words, not in the programming language. This stage trains students to think about the direction and approach such that students can explain their idea clearly in interviews and exams. Secondly, the student can attempt to solve the coding challenge programmatically. After the student passes all test cases, it will automatically create a new post to the forum for reviewing by others and get some experience points on “Adventurer” and sill. If the student does not have life left, the page will refresh automatically after five seconds.

![Diagram](image)

**Figure 3.6: Flow of the coding challenge**

### 3.4.3 Coding Contests and Courses

The coding contests are the same as LeetCode, which provides several coding challenges to solve directly. Students are required to solve the questions within the time limits. In the end, students are ranked by their time used.
Students can learn how to use different programming languages and methods. The project team decided to use multiple choice for simplicity. After finishing all questions, they should solve the specific coding challenge(s). During the course, students can see their progress on the course. In the end, the course will reward students with a badge.

### 3.5 Comparison to Existing Websites

Table 3.2 shows the comparison between this project and existing websites. Compared to LeetCode and HackerRank, this project focuses on education for university students only, not interview practice for any students. This website also restricts access to some posts mentioned in Chapter 3.3. For a session of attempting a coding challenge, this website has a limited chance of submission and contains several steps, while the existing websites have unlimited access to the problems and public access for all posts. Furthermore, courses in LeetCode only have a list of questions for users to finish, but this project website also teaches users how to use different programming languages and algorithms.

<table>
<thead>
<tr>
<th></th>
<th>This Website</th>
<th>LeetCode &amp; HackerRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Education</td>
<td>Interview practicing</td>
</tr>
<tr>
<td>Target User</td>
<td>Students</td>
<td>Any users</td>
</tr>
<tr>
<td>Solving a coding challenge</td>
<td>several steps</td>
<td>direct submission</td>
</tr>
<tr>
<td>Given chance in a session of coding challenge</td>
<td>Limited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Restricted access to some posts</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Courses</td>
<td>Teach and attempt</td>
<td>Attempt directly</td>
</tr>
</tbody>
</table>

Table 3.2: Brief comparison between this website and LeetCode and HackerRank
Progress Status

This chapter describes the details of the progress of the project. Overall, the project team finished designing the essential pages in Figma and some pages. There are difficulties in the development process, but using open-source systems can reduce the complexity. The project team also focuses on the scalability, usability, and customizability of the website or systems.

4.1 Current Progress

4.1.1 User Interface Design

Currently, the project team designed the pages of the website in Figma. The designs referenced other existing products, including Reddit, StackOverflow, HKU Moodle, and LeetCode. Appendix C lists the available User Interface designs.

4.1.2 Non-game contexts

Currently, the project team implemented the Entity Relation Diagram for the challenge, login, and forum pages. The project team then converted the diagram into collection types in the CMS. Based on the design, the project team implemented the following pages: a user sign-in page, a user registering page, a forum page, a post details page, a challenge list page, and a challenge details page.
4.1.3 Coding Challenge

The project team implemented a function name “Judger” to handle the judgment of submission. Figure 4.1 shows how Judger handles it. Firstly, the Judger will get the results from CMS. For every test case, the Judger will take the stdin and post a new submission to Judge0. In the end, Judger will return the result details by result or comparing the stdout from the student and the expected stdin.

![Lifeline diagram of coding challenge](image)

Figure 4.1: Lifeline diagram of coding challenge

4.1.4 React Hooks for gamification

As mentioned in the previous chapter, there are two steps in a coding challenge. The project team implemented several react hooks to achieve the workflow, and the project team designed “usePlayer” and “useMonster” hooks for gamification (see Figure 4.2 and 4.3 for their construction). The idea is similar to some real-life games. Monsters and players have their own life points, and actions, and may have changes in life points. The hook will call “onLifeChange” for this situation. When the player or monster has no life remaining, it handles what to do in the “onDead” function. A monster may have
several stages, and “useMonster” also manages the stage.

Figure 4.2: The state, parameter and functions in “usePlayer”

Figure 4.3: The state, parameter and functions in “useMonster”

They have a “worker” function that performs an action, attacking, for example. Figure 4.4 shows the flow of the worker given the user inputted the parameters in the hook. In the beginning, the worker initializes the work in the “beforeWorking” function (For example, creating a new instance). Then the worker runs “work” function. After that, the worker also does some work in “afterWorking” function, updating the database, for example. Given the result from “beforeWork” function, “work” function, and “afterWork” function, the worker can check whether the work is successful work or not in the “isSuccess” function. If it is a successful work, the worker will run the “onSuccess” function, “onFail” function otherwise. For example, the website can show the dialog when successful. Users can close some work in “onEnd” function. For example, users can disconnect the database in this function. (Appendix D shows a detailed example.)
4.2 Project schedule

The project team designed and implemented some pages. The project team connected the CMS and Remote Code Execution System to the front end. The project team is adding and creating the points and leveling system for existing pages. After that, the project team will work on the user profile and leaderboard page. Lastly, the project team will work on the course and contest pages.

4.3 Difficulties

This project faces difficulties in implementation. There is only one person on the project team, developing all services from scratch is difficult. Therefore, this project
looks for existing services to reduce the complexity and time of development. For example, this project uses Strapi as the CMS and Judge0 as the Remote Code Execution System instead of developing it from scratch. The project team also focuses on the scalability, usability, and customizability of the website or systems. For example, how the project team creates the “useMonster” hook. Moreover, Judge0 allows different programming languages for students. However, the project team restricted it to Python only, and students cannot create a new coding challenge because of time limits and difficulties in handling the input of (complex) data types.
Chapter 5

Conclusion

Currently, the e-learning environment of CS in HK is not motivating. The few discussion, interactions, and motivation make students less competitive than others. This project designs a gamified website for CS students to engage them in learning. However, this website focus on programming, an essential subset of CS, given the time limit. The project team uses Strapi and Judge0 as the CMS and Remote Code Execution system, which interact with Next.js with the middlewares, Redux and Next-Auth.

The project team finished the User Interface designs in Figma. Based on that, the project team implemented the user login, user register, coding challenge, and forum pages. Currently, the project team is adding the points and leveling systems to the websites. After that, the project team will work on other pages, including the user profile, leaderboard coding contests, and course pages.

There are some limitations to the website. As mentioned in Chapter 1 and Chapter 3, this project may implement functions with AI or ML for the point system. However, the project team will not tune and research for the correctness of the model because of the time limit. For the coding challenge, the project team only allows the programming language to be Python. Furthermore, the template of coding challenges is non-auto-generated. Because of the time limit, the project team cannot implement an efficient way to input test cases and their (complex) data types. Therefore, students cannot create a new challenge currently.

Furthermore, experiments in studies show that gamification can possibly motivate students to learn with gamification. However, this project will not research the effectiveness of the gamified website because it takes a long and continuous time.
When the project team finishes this project, researchers can use it as the prototype for university learning. Researchers can also evaluate its effectiveness and possible changes through experiments. For example, future experiments can invite some students to use the website and make evaluations by comparing it with traditional learning.
Bibliography


Appendices
The following are the three section of gamification design. [4]

- **Mechanics**: Points and badges that concern data representation and algorithms.

- **Dynamics**: Players’ inputs and outputs over time are concerns of the runtime behavior of mechanics like completion and choices.

- **Aesthetics**: Users’ emotional responses after they interact with the gamified system. For example, the feeling of being challenged.
Self-determination theory

The following are the three sections in self-determination theory [1],[5].

- **Relatedness**: Interactions and connections with others, their goals, their interests.

- **Competence**: Challenging tasks aiming to complete their goals are what personal interests in.

- **Autonomy**: It satisfies the need to control one’s own life by participating in voluntary play to fulfill a person’s personal goals.
Page Designs

The following are the designs on important pages of the website. There are comments that indicate the need to change the pages. The gray rectangles are images that can be replaced in future development.

(a) Home Page of the website
(b) Post List Page of the website
(c) Post Details of the website

(d) Login Page of the website
(e) Register Page of the website

(f) Add Post Page of the website
(g) Challenge Page of the website

(h) LeaderBoard Page of the website
(i) Contest list Page of the website

(j) Contest Page of the website

(k) Course List Page of the website
(1) User Profile Page of the website

Figure C.1: Pages of the website
Example of worker function

Take the Coding Challenge as an example. A user is a player to kill the monster (Coding Challenge). The worker function performs as follows.

1. “beforeWorking”: Do nothing because there is no initial work.
2. “work”: Call the “handleCompile” function.
3. “afterWork”: Submit a new attempt to the CMS.
4. “isSuccess”: Whether the work result is a success or not. Then run “onSuccess” when “isSuccess” is true, “onFail” otherwise.
   (a) “onSuccess”: Show a success dialog, create a new post, and give experience points to the user; Reduce monster life points by one.
   (b) “onFail”: Reduce player life points by one
5. “onEnd”: Do nothing because there is no remaining work.

For the life points of the player:

- When a player has change in life points (“onLifeChange”), the website shows a toast to user (inform the player that the submission does not pass all test cases).
- When a player has no life points (“onDead”), the website shows a dialog to user (inform the player that the he failed to kill the monster) and refresh the page after five seconds.

For the life points of the monster:
• When a monster has no life points ("onDead"), it drops experience points to user and update the experience points in the CMS.