

Palestine Technical University - Kadoorie

Faculty of Engineering and Technology

Computer Systems Engineering Department

**GAMIFYING PROGRAMMING LEARNING AND PROBLEM SOLVING**

**Prepared by:**

Majd Kittaeh

Omar Shaikh Ibrahim

**Supervised by:**

Nael Salman, Ph.D.

**A graduation project submitted in partial fulfillment of the requirements for the bachelor's degree in computer systems engineering.**

Tulkarm, Palestine

January, 2023

ACKNOWLEDGEMENT

We begin by expressing our gratitude to Allah for His blessings and guidance throughout this journey. We would like to extend our heartfelt appreciation to our parents and families for their unwavering support and encouragement that enabled us to successfully complete this work. We are also deeply grateful to our university Palestine Technical University - Kadoori and specifically the Faculty of Engineering and Technology for providing us with the opportunity to pursue our studies and engage in this project.

Our utmost appreciation goes to our supervisor, Dr. Nael Salman, for his invaluable guidance, continuous support, and dedication. His expertise and insightful feedback have greatly contributed to the success of this project. We would also like to extend our sincere thanks to all the faculty members of the Department of Computer Systems Engineering for their valuable teachings and assistance, which have been instrumental in shaping our knowledge and skills.

Lastly, we would like to acknowledge the efforts of everyone involved in this project, whose contributions have been instrumental in its completion. We are grateful for their collaboration and support throughout this endeavor.

**ABSTRACT**

This thesis presents the concept of fusing game design with programming teaching, aiming to leverage the principles of game design to create an engaging and effective environment for individuals to learn programming concepts.

The idea revolves around using game mechanics, such as levels, exploration, puzzles and progression to change the challenging process of programming into an interactive and enjoyable journey.

This conceptual project sets the stage for further exploration and refinement. It raises questions about the good integration of game mechanics, choosing languages and the design of challenges to maintain a balanced blend of fun and education.

As a conceptual project, it will represent an innovative and forward-thinking approach to address the challenges in programming education. As the concept evolves, it holds the potential to redefine how individuals approach and engage with programming learning.

**CHAPTER 1**

INTRODUCTION

*In this chapter, we will discuss the outlines of our project.*

**1.1 Background**

With the rise of generations that mainly consume digital content especially visual, there is a shift happening in learning preferences towards interactive and dynamic approaches. The traditional lecture-based learning may not fully capture the attention and interest of those individuals.

At the same time in this digital era, the ability to code is not only valuable for dedicated software developers but is increasingly becoming important for professionals in different fields.

Video games and programming share the same core : problem solving. Recognizing this opens up opportunities to leverage the interactive nature of gaming for educational purposes.

Another thing that supports the project direction is the scene of games modding –short for modification- communities, where players who are often teenagers learn programming to modify a game and change its content, which shows that players in high numbers are motivated to learn programming because of games.

**1.2 Problem Statement**

Many programming courses focus heavily on theoretical concepts, which gives learners no opportunities to apply their knowledge to real-world problems. Creating a video game environment with its own rules and objectives simulates the situation of writing a code to solve a real problem.

Traditional education follows a one-size-fits-all approach, which makes a problem for individuals with different learning styles, this and what we mentioned in the previous section about the newer generations being digital content consumers weakens the classroom traditional setting and calls for a more interactive methods.

**1.3 Objectives**

1. To develop a conceptual project that investigates the idea of fusing game design and programming teaching, and to provide a framework for guiding the development of such platform.
2. To ensure a smooth learning curve by starting with basics and gradually advancing and giving more difficult challenges.
3. To use challenging nature of both video games and programming to create a satisfying experience that ends with a sense of accomplishment.

**1.4 Procedures**

To achieve our project goals, we focused on developing a computer game that allows users to solve puzzles and proceed in levels by writing the proper code using a custom programming language made for our project.

To be able to do this, we started learning about interpreters and how to make a programming language in addition to studying some game design basics. Thus, we gathered the information we needed to start designing.

After gathering requirements, we analyzed and organized the information using UML (Unified Modeling Language) diagrams. These diagrams represented the structure and behavior of each functionality in the system (both game and interpreter), providing a blueprint for how they would be implemented and integrated.

Once the design phase was completed, our team transitioned into the development phase. For this, we adopted the agile approach to the Software Development Life Cycle (SDLC). The agile model is widely recognized in the software development industry, as it promotes flexibility and iterative development strategies. Our workflow progressed through cycles of planning, designing, coding, and testing specific sets of features or functionality.

By adopting the agile methodology, we were able to ensure regular communication and collaboration within our development team. This facilitated efficient decision-making, effective problem-solving, and continuous feedback loops.

Throughout the development process, we prioritized regular testing to verify the functionality, performance, and usability of the system. This iterative approach allowed us to identify and resolve issues early, ensuring a smooth and reliable user experience.

**1.5 Organization of the Study**

This thesis is divided into five chapters each one describing a part of our project.

The chapters are as the following:

* Chapter 1: Introduction: This chapter outlines the idea and the main objective of the project.
* Chapter 2: Literature review: This chapter discusses similar previous projects.
* Chapter 3: System Requirements: This chapter discusses the System Requirements (functional and non-functional).
* Chapter 4: Methodology And Technologies: This chapter discusses the methodology we used during the construction of this project.
* Chapter 5: This chapter reviews the conclusion and Future Vision for this project.
* Chapter 6: References.

**CHAPTER 2**

LITERATURE REVIEW

*In this chapter, we will discuss similar previous systems*

**2.1 Similar Work and Differences**

Scratch [0.0] is a visual programming language and online community that allows users to create interactive stories, games and animations using its simple lego-style programming language. Scratch is similar to our project as it focuses on the same audience of young students and newcomers to programming. However, it does not offer the code-to-win concept or adding game elements to the teaching process.

CodinGame [0.0] is an online platform that offers coding puzzles and challenges in the form of games where users can solve programming problems using various languages and compete with other programmers. CodinGame is similar to our project as it offers the code-to-win concept and uses puzzles and levels to teach programming. However, it uses established programming languages which does not give the flexibility we want to achieve in our project. In addition to that, a big difference is that it lacks the real-time feedback and interactivity as it plays the winning scene if the code is true and the losing one if it is false, it does not provide enough game elements.

Codecademy [0.0] is an online learning platform that offers coding lessons in various programming languages where users learn by writing and running code directly in the browser. It is similar to our project with its use of level and progressive learning but it lacks the interactivity and game elements.

**CHAPTER 3**

SYSTEM REQUIREMENTS

*This chapter discusses the System Requirements (functional and non-functional).*

**3.1 Functional Requirements**