INTERNATIONAL UNIVERSITY

VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY

School of Computer Science and Engineering

-----*-----**

PROJECT REPORT

DRAGON TALE

Advisor: Dr Tran Thanh Tung and MSc Nguyen Quang Phu

Course: Object – Oriented Programming

No.	Full Name	Student's ID	Contribution
1	Phạm Trung Tín	ITDSIU19019	25%
2	Phan Bảo Trân	ITDSIU21125	25%
3	Lê Huỳnh Nhã Nguyên	ITDSIU21058	25%
4	Nguyễn Đức Nguyên Phúc	ITDSIU21108	25%

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Dragon Tale is an adventurous 2D game where players control a dragon navigating through perilous terrains, battling diverse adversaries, and collecting coins to amass points. The primary objective is to eliminate monsters and gather as many coins as possible, enhancing the player's score.

Dragon Tale combines strategic combat, agile movement, and prudent decision-making, offering players an immersive experience as they navigate dangers, accumulate points, and triumph against a variety of formidable foes. In this commitment, our team has developed the game "Dragon Tale" to provide players with a version suitable for all ages. The theme of the game is based on the idea of the adventures of a Dragon within nature. This is the product of our efforts to offer players, often young students, an entertaining game.

Keywords: dragon tale, adventurous, game 2D, object-oriented programming.

CHAPTER 1: INTRODUCTION

1. Objectives

The project goal is to create a game based on 2D game concepts and concepts ofa platform game. The game demonstrates core ideas of object-oriented programming and some fundamental design patterns. Besides, it is a standalone game that doesn't demand exceptional skills or sophisticated technology. This simple yet engaging game is suitable for individuals of all age groups. Dragon Tale presents an exciting 2D adventure where players take charge of a dragon manoeuvring through dangerous landscapes, engaging in battles with various foes, and collecting coins toaccumulate points. The main goal is to defeat monsters and gather coins to increasethe player's score. Through our efforts, we've designed a user-friendly program aimed at providing users with the most enjoyable experiences possible. Furthermore, evaluate the skills to build new features on top of the existing ones.

To be short, the project aims to:

- Developing a game brings entertainment to players.
- Applying Object-Oriented Programming (OOP) techniques within the Theory class.
- Go through the process of game management and enhancing the code.
- Assessing the potential for expanding the basic code by incorporating additional features.

Figure 1. Dragon Tale banner

2. The tools used

- IDE for programming and debugging: IntelliJ.
- Java Development Kit: 21.
- Mean of code version management: GitHub.
- Means of contacting: Facebook and Discord.

Figure 2. GitHub statistics

CHAPTER 2: METHODOLOGY

1. Rules

The following is an overview of the key points about the rules of our game.

- **GAME AIMS** to slay every monster and collect coins along the way to obtain the greatest number of points and coins possible.
- **BEGINNING PLAYER** begins with 10 health, 25 fireballs that reload after firing, and no cash or points.
- **THE PLAYER** battles against various adversaries, known as small monsters and at the end of the game, a large monster will appear. Users will control the Dragon.
- **THE INTERACTIVE TO:** combat against a variety of monsters, such as Moneys, Birds, Sluggers, Heroes, danger zones, and a boss monster (Spider). Especially during the gameplay, it is possible to collect money.
- **PASSING STARTING POINT**: The locations where you start are extremely dangerous and teeming with creatures.
- **PLAYER**: You are the dragon's embodiment, and you can manipulate the one with keys.
- **KEY**: Along with other keys like W for jumping, E for gliding, R for scratching, and F for firing, KEYS includes ⇦ , ⇧, ⇩ , ⇨ for the movement.
- **FIREBALLS**: The dragon's fireballs, which are generated by player bullets, shoot when you hit the F key.
- **FIRING AND SCRATCHING:** There are two ways for a player to attack: firing for long-range strikes and scratching for short-range attacks.
- **ENEMIES**: Besides avoiding or travelling through the teleports, every monster has a different amount of poison and damage.
- **ARACHNIK**: is a tiny spider which moves up or down in response to harm, and the amount of damage is one.
- **HATMONKEY AND SLUGGER** are two creatures that move from right to left and go back with the damage of two.

- **BIRD** is a robot disguised as a bird that soars at a height, dealing 4 damage and requiring a jump to destroy.
- **HERO** you must use caution when handling them because their damage is three.
- **BOSS** is a king spider that is excessively big for any object on the map. It shoots venom that deals damage equivalent to a mushroom. You should avoid it since every time you cross paths with it, you will lose five health.
- **VENOM**: A Boss's contaminate is referred to as a "venom," and it is released from the spider.
- **TELEPORT**: You must stay away from them since it's a dangerous spot where you will perish if you traverse it.
- **COINS AND POINTS**: You get an extra coin and 100 points for eliminating an adversary. These are the same coins that are found on the streets.

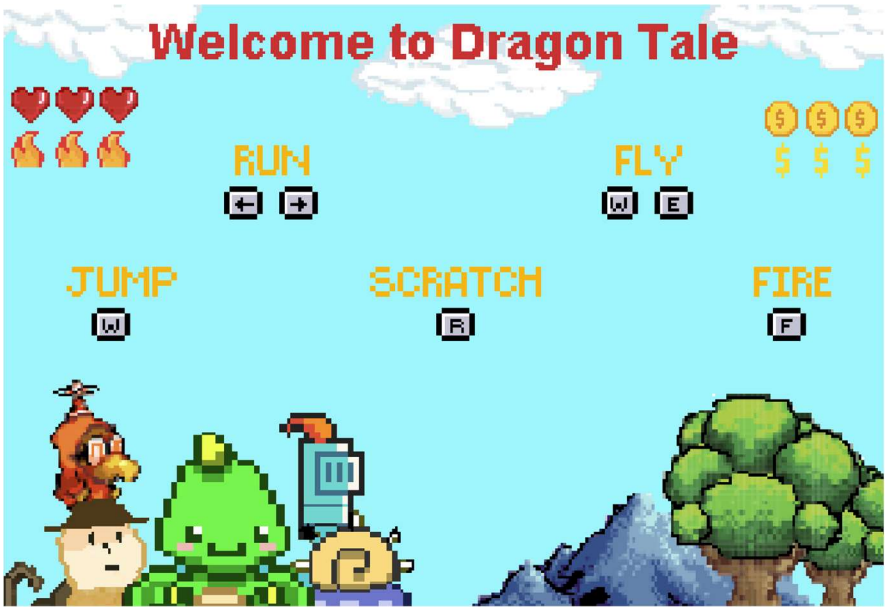


Figure 3. Help screen

2. Design

a. UI/UX

Since UI/UX has always been our focus before we launched the game, we have received a lot of input from other people to establish a pattern.

A 2D adventure and action game is called Dragon Tale. The game narrates the tale of Dragon Tale, a green dragon who is courageous and strong yet also good-natured and giving.

In the game, players will battle monsters by controlling Dragon Tale. The game is appealing to players of all ages and incorporates gorgeous 2D graphics.
The following crucial components have been identified to build the UI/UX for the Dragon Tale game:

- Information to display:
    - Player

Figure 4. Player

    - Mini monsters

Bird

Hatmonkey

Slugger

Arachnik

Hero

Figure 5. Mini monsters

- Boss

Figure 6. Boss

- Health

Figure 7. Health

- Fireball

Figure 8. Fireball

- Coin

Figure 9. Coin

- Venom

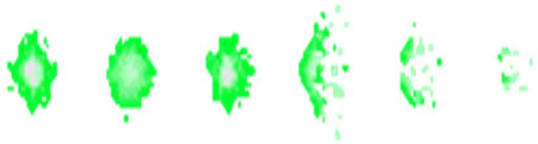


Figure 10. Venom

- Interaction needs to be provided.
  - Control the character

Key	Action
⇦ ⇨	Run
W	Jump
R	Scratching
F	Firing
W+E	Fly

- Use ⇦⇨ to move the dragon.
- Use R and F to attack the enemies.
- Watch the amount of fireballs below your amount of health as it decreases when you use F.
- Fight the boss must avoid touching it and dodge its venom.
- Should be based on the rugged terrain to kill the boss.

b. Game algorithm

After much iteration and troubleshooting, we now have the project structure asseen in the image below.

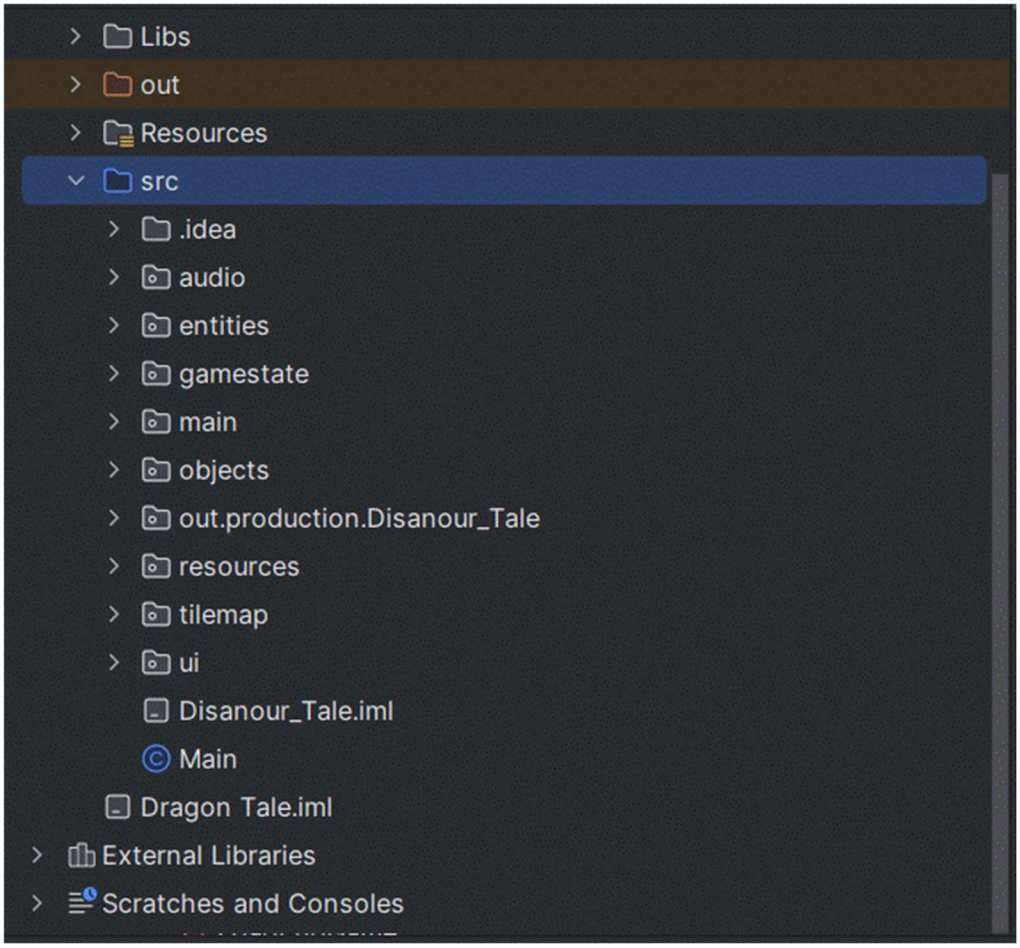


Figure 11. Project structure

The classes we teach can be arranged into distinct groups, like:

- MapObject: is the superclass of the classes including the Player class to display the player's information (dragon tale), the Enemies class to describe various monsters in the road, the teleport class, the venom class to describe the enemies' venom to attack the player, and the collectable class to describe the coins in the journey.
- Enemies: is the superclass of all the classes, which includes the boss category or classes that are similar to mini monsters like Hero, Slugger, Arachnik, and Hat Monkey.

- Die Enemies is the superclass of all the classes, which includes DieBoss, DieHero, DieSlugger, DieArachnik, and DieHatMonkey.
- Animation: is used to describe the frame's actions of Entities.
- HUD: displays information including fireballs, health, coins, and scores.
- Background: displays the background of the game
- Tile and TileMap: are used to display deep holes and walls in the game.
- AudioPlayer: creates the music, even the sound of Dragon's firing and scratching.
- Abstract class GameState displays the state of the game; its subclass HelpState demonstrates how to play; Level1State depicts the initial fight screen; MenuState shows several options; and WinnerState and GameOverState indicate when the player succeeds or fails. GameStateManager: is manager class various the state class inherits GameState class.
- Game and GamePanel: is initiated even controlling the programing.

3. UML Diagram

We provided the UML diagrams for the entire project and each group that was addressed to help you better understand the structure and methods.

- Whole UML Diagram: OOP-DragonTale-Game/UML/UML.pdf at main · nguyenducnguyenphuc2002/OOP-DragonTale-Game (github.com), access to view the whole UML Diagram.

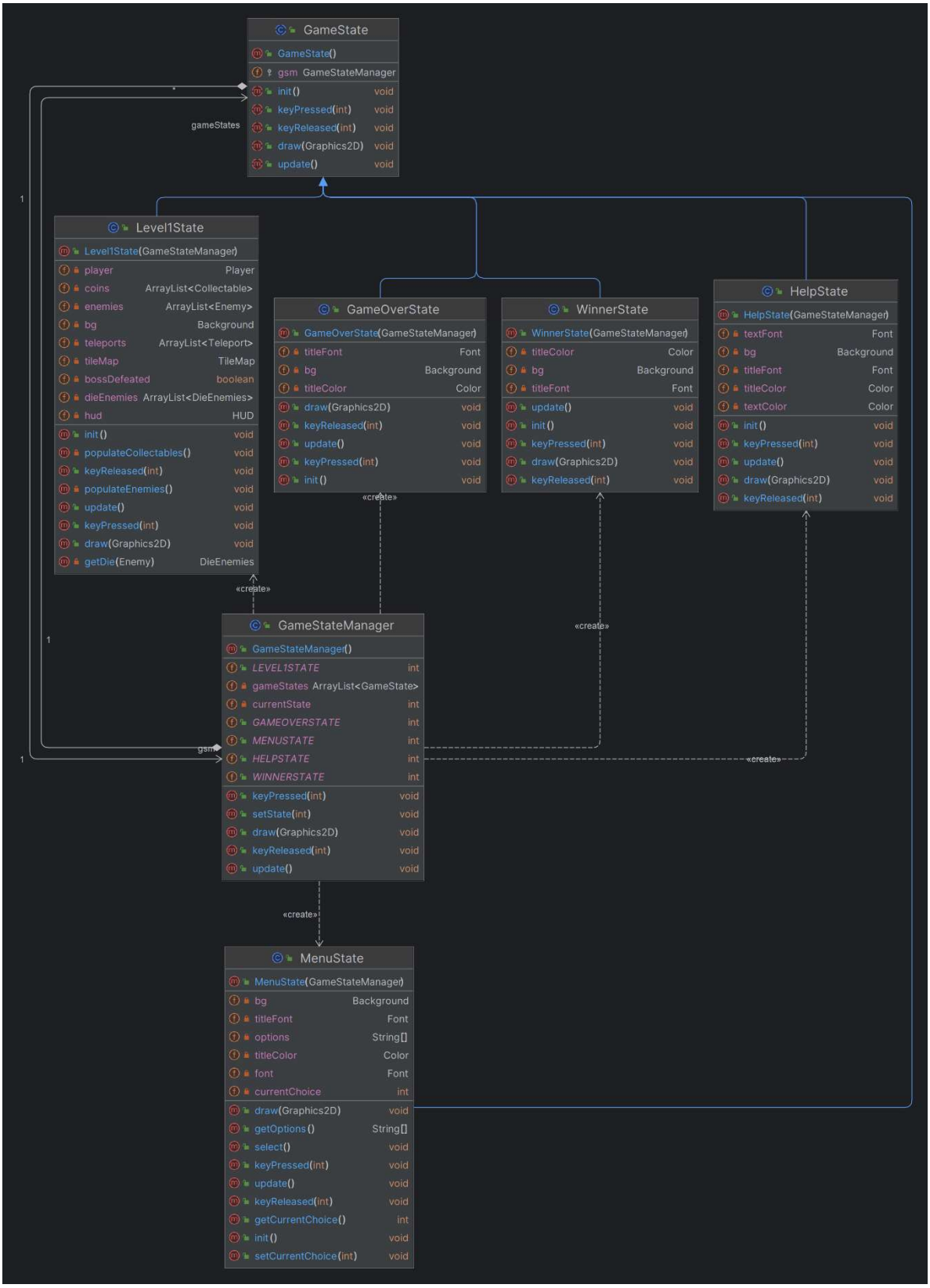


Figure 12. GameState diagram

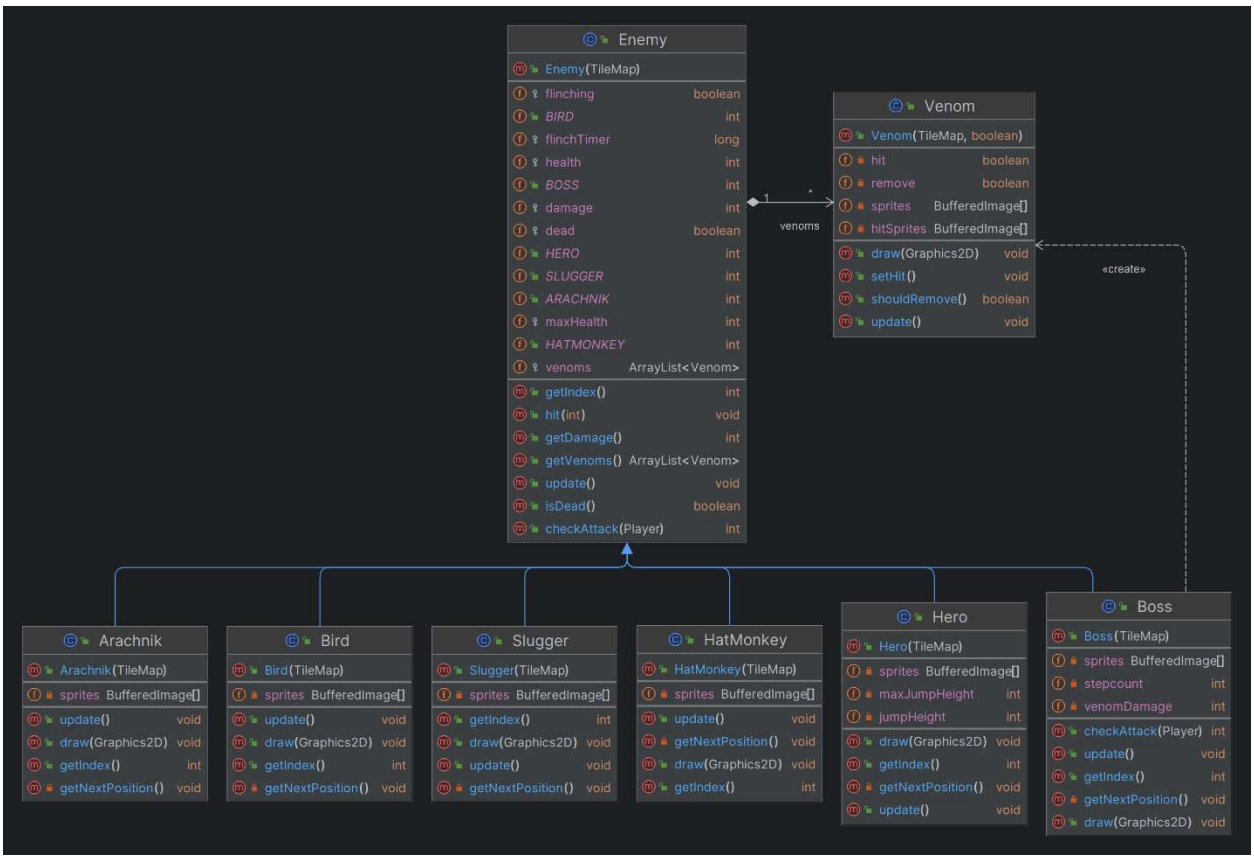


Figure 13. Enemy diagram

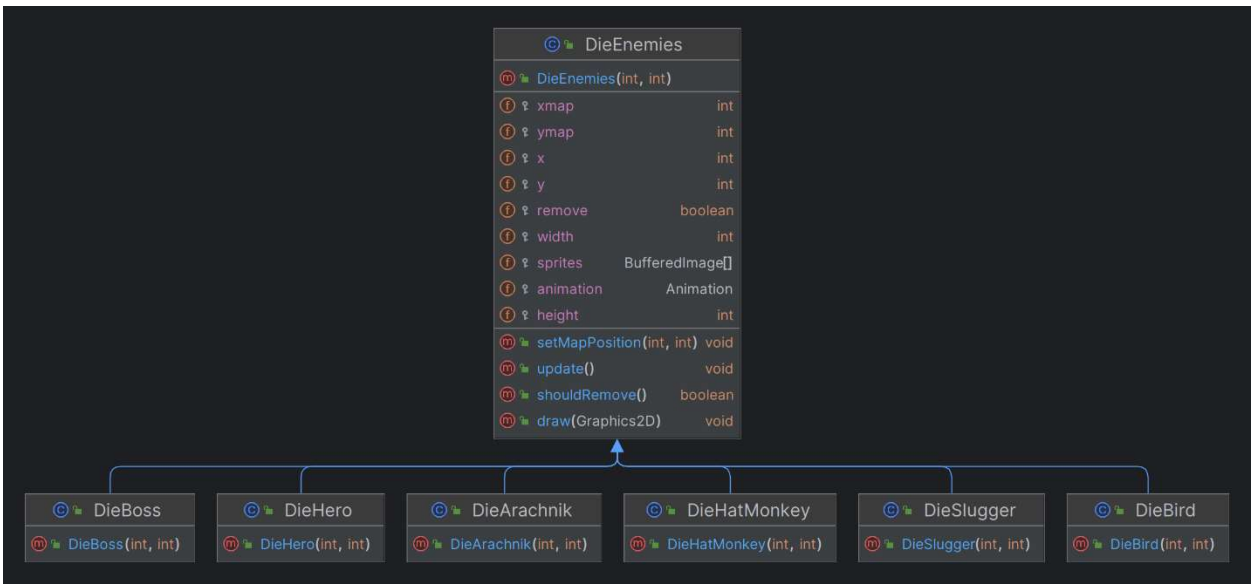


Figure 14. DieEnemies diagram

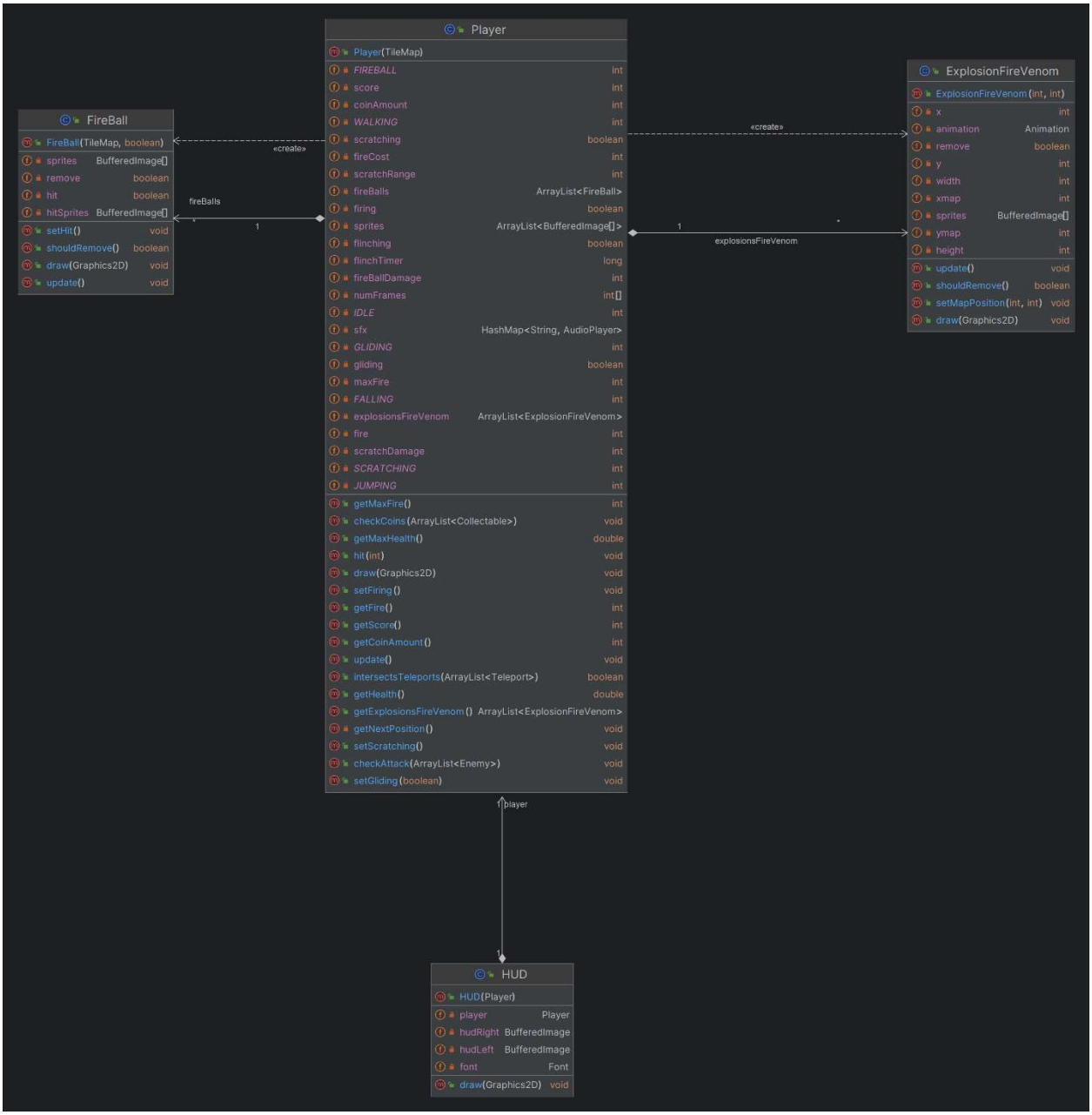


Figure 15. Player diagram

Figure 16. Objects diagram

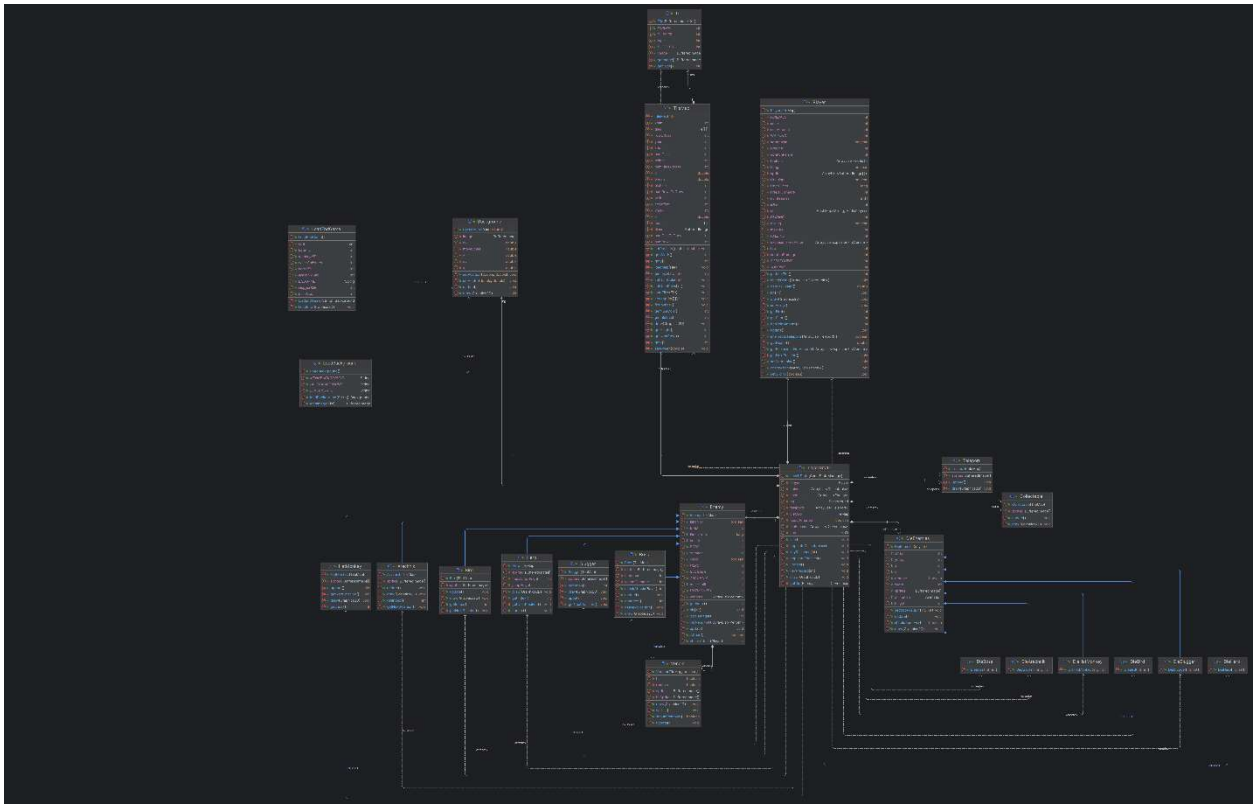


Figure 17. Level diagram

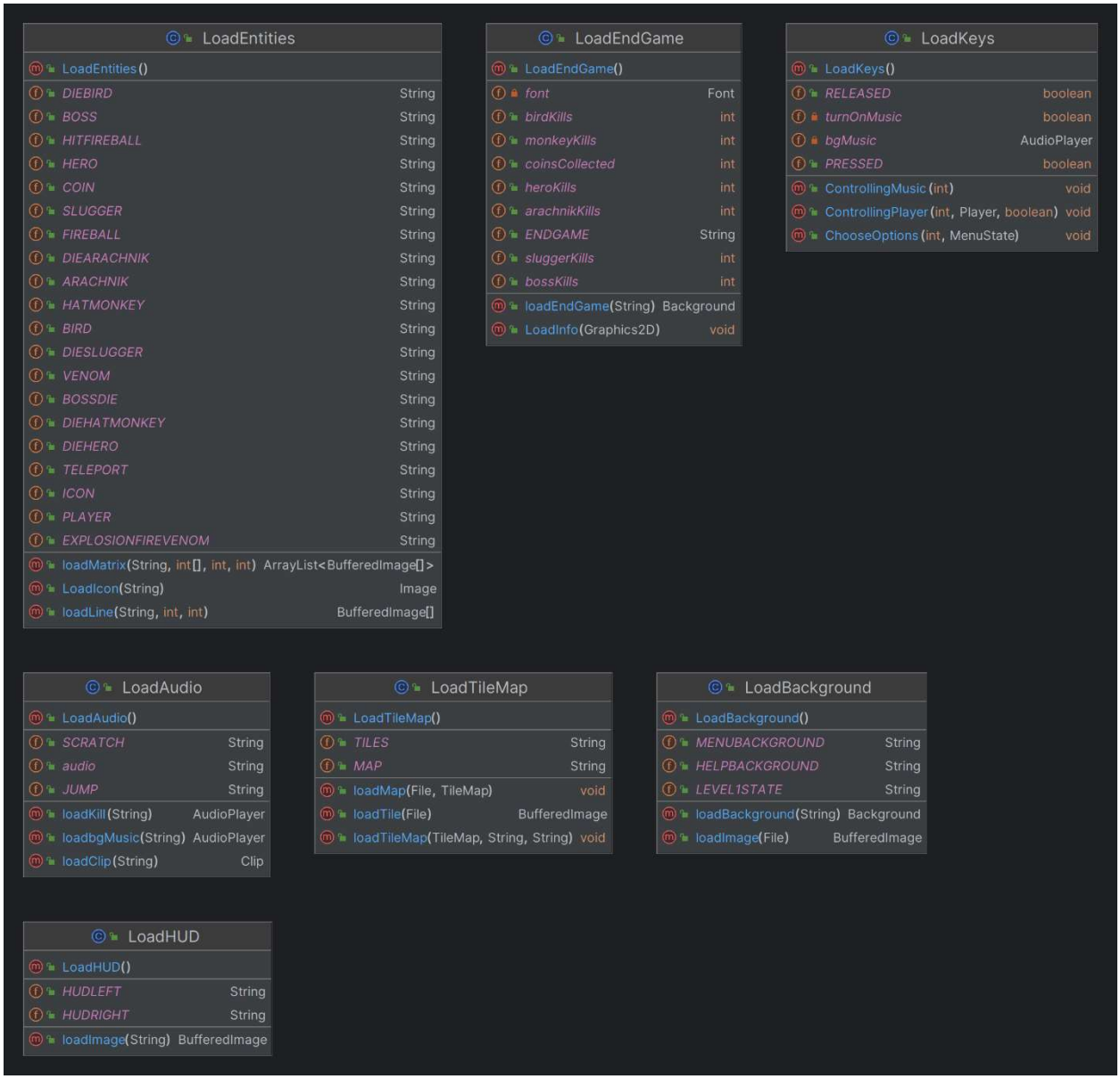


Figure 18. Ui diagram

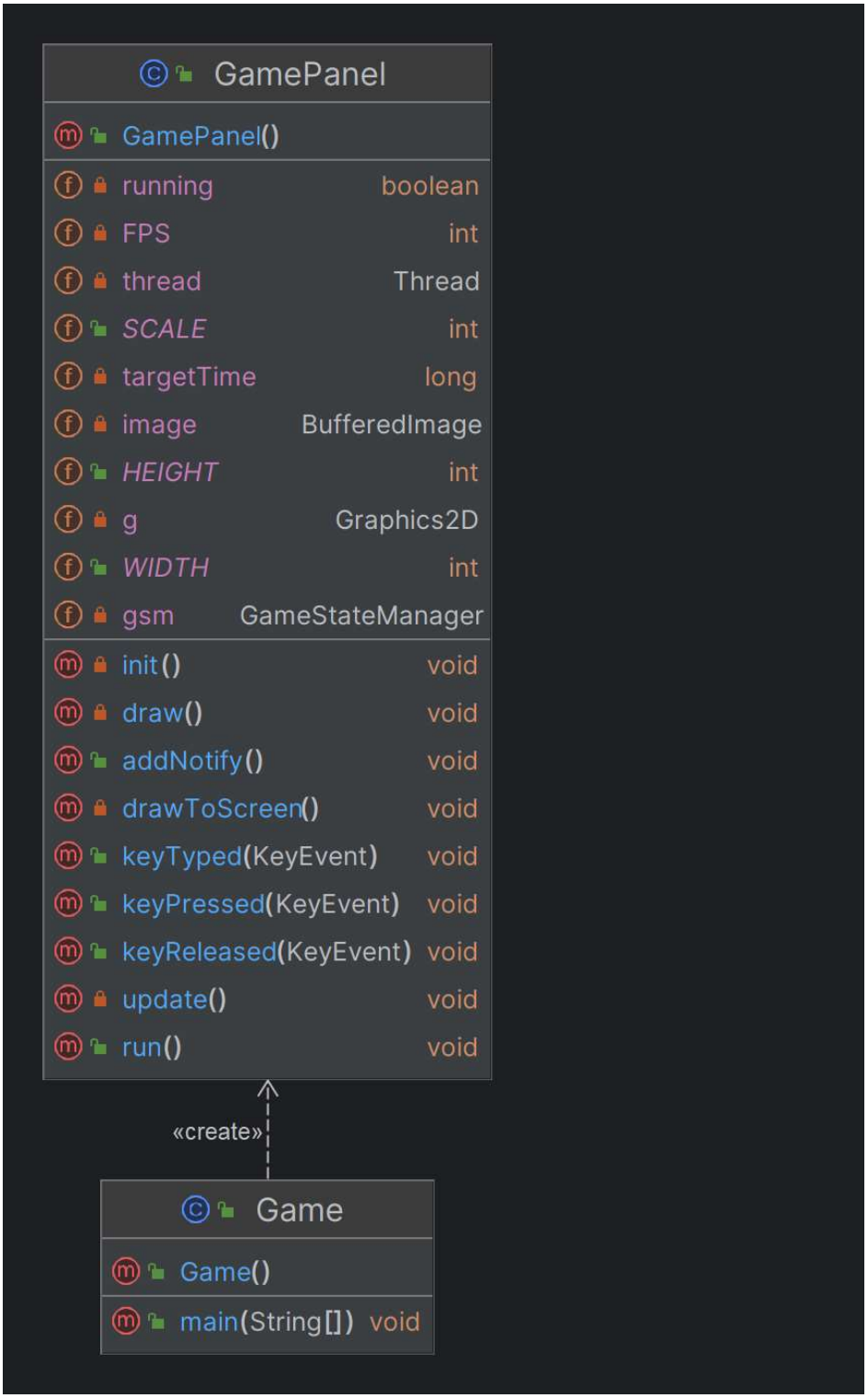


Figure 19. Main diagram

CHAPTER 3: DEMO – RESULT

To test our game on a machine that had an IDE and Java Development Kit 21 installed.We pulled our git repository and ran the Main class to launch the game. The following screenshots show some examples of the game states in the current build.

Figure 20. Menu screen from the game

Figure 21. Help screen

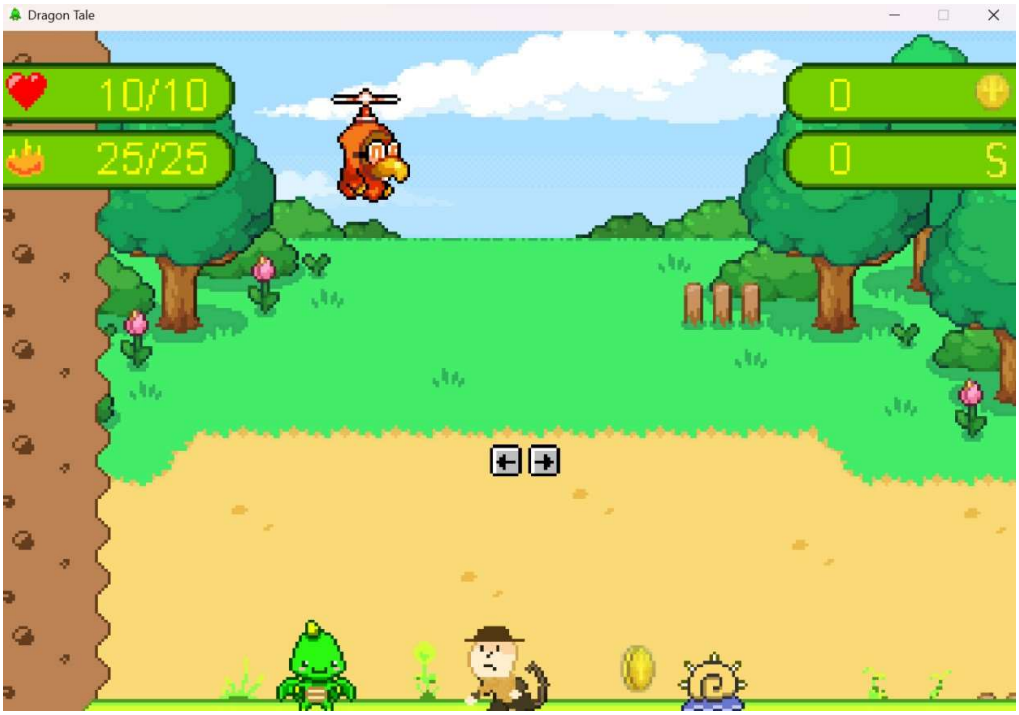
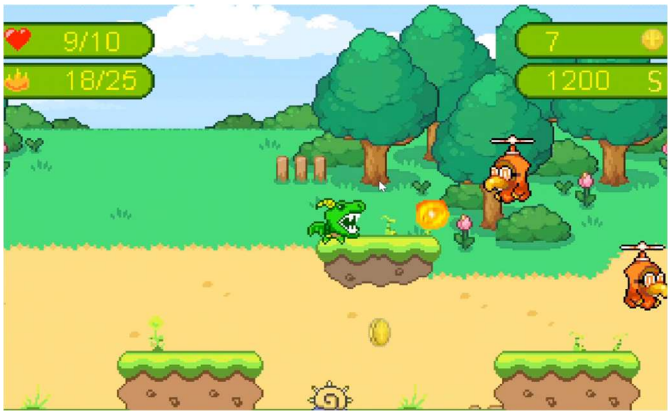


Figure 22. The start of the game



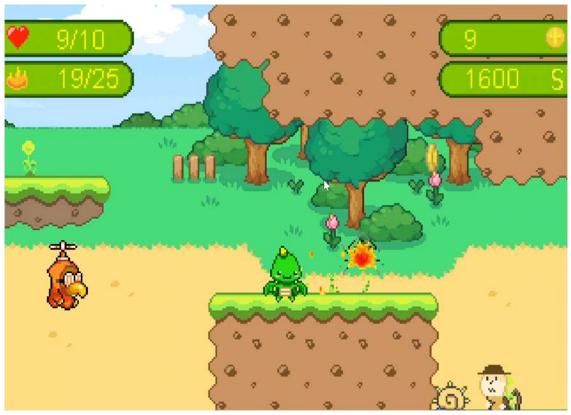


Figure 23. During the game play

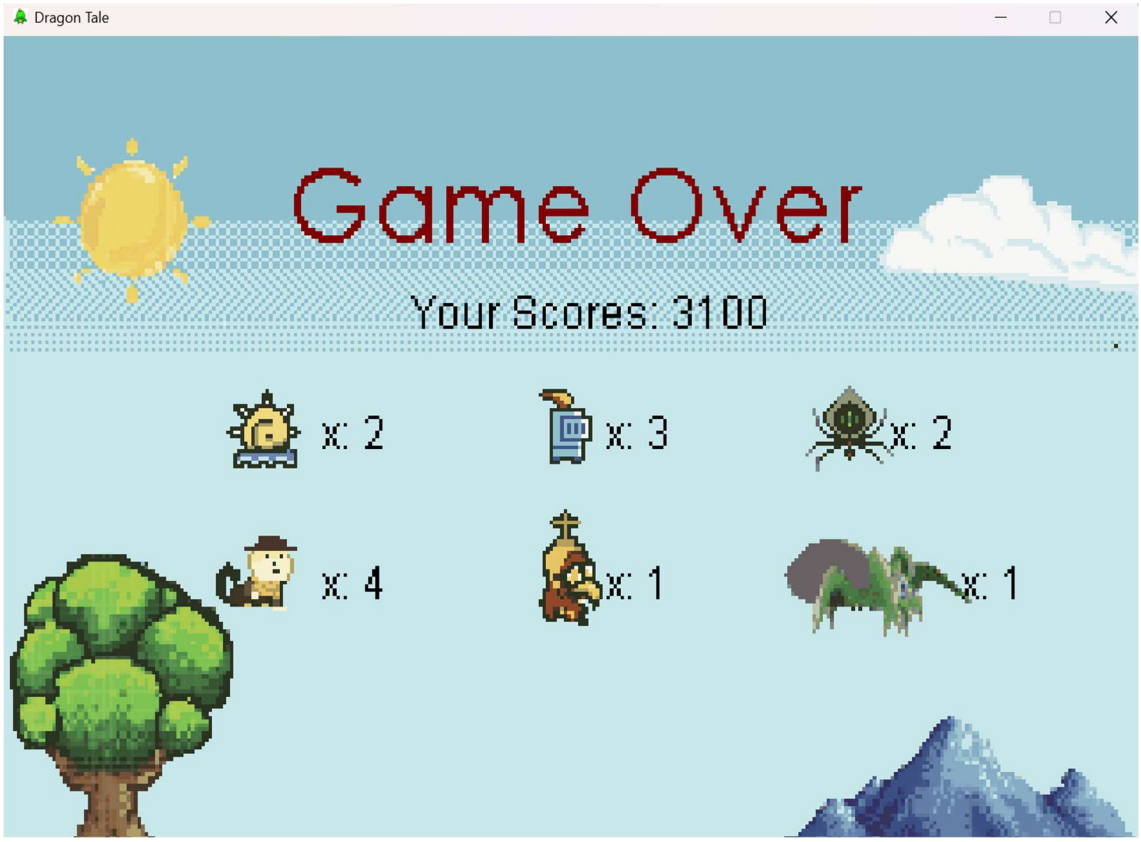


Figure 24. End the game

CHAPTER 4: CONCLUSION AND FUTURE WORKS

1. Conclusion

The game's development has not yet ended. In the final term, the team has acquired a deeper understanding of the four fundamental features of Object-Oriented Programming (OOP) and the SOLID principles. This knowledge has been instrumental in enhancing our proficiency in OOP within the realm of game development as well as in the programming process after finishing a game with some novel features compared to the original version. The project's classes have discussed the concept of encapsulation. Objects belong to enemies' package and player package is where inheritance, abstraction, and polymorphism have been used the most frequently. Because of this, Dragon Tale was strictly developed using the fundamental idea of OOP, and the game code contains all four key OOP features and a design pattern learned from class. As we approach the conclusion of the project, it is crucial to acknowledge that these team members will not be afforded a second opportunity to work on this. The wealth of knowledge gained in this endeavor is not only a testament to our collective expertise but also a testament to the innovative spirit that has driven us to push the boundaries of game development.

2. Future works

Unfortunately, the team was hoping to develop power-up items that players can collect during gameplay. These power-ups could enhance the player's abilities, offer temporary invincibility, or provide special attacks, adding an extra layer of strategy to the game. In addition, expand the gaming experience by introducing additional levels with distinct challenges, environments, and enemy types, design levels with a mix of terrains, including platforms, cliffs, and water bodies. Varying terrains will require players to adapt their movement strategies, adding an element of exploration and agility to the gameplay. Furthermore, incorporating a feature that allows players to customize their characters with various skins, outfits, and accessories will also be applied in future updates, players can tailor their characters to reflect their individual style preferences, creating a more immersive and enjoyable gameplay journey. Therefore, any new commitments are highly appreciated.

3. Acknowledgment

We would like to convey our deepest appreciation to our instructor and individuals who assisted us in reaching the goals of this project:

- Dr. Tran Thanh Tung
- MSc. Nguyen Quang Phu
- Original code from foreignguymike (Foreignguymike, 2018)
- The sites Geeksforgeeks, Javapoints, and so on
- The README.md template from othneildrew (Drew, 2018/2022)

REFERENCES

[1] ForeignGuyMike. (2018). foreignguymike/dragontale [Java]. Retrieved December 10, 2023, from https://github.com/foreignguymike/legacyYTtutorials/tree/master/dragontale

[2] Java Swing Tutorial. Javatpoint. (n.d.). Retrieved December 11, 2023, from https://www.javatpoint.com/java-swing

[3] UML Diagrams Tutorial. Javatpoint. (n.d.). Retrieved January 4, 2023, from https://www.javatpoint.com/uml-diagrams

[4] Drew, O. (2022). Othneildrew/Best-README-Template. Retrieved January 4, 2023, from https://github.com/othneildrew/Best-README-Template

[5] SOLID Principle in Programming: Understand With Real Life Examples. GeeksforGeeks. Retrieved December 28, 2023, from https://www.geeksforgeeks.org/solid-principle-in-programming-understand-with-real-life-examples/

[6] Playing music in a Java game. stackoverflow. Retrieved December 16, 2023, from https://stackoverflow.com/questions/27854171/playing-music-in-a-java-game

[7] Ileriayo Adebiyi. (2020). Ileriayo/ markdown-badges. Retrieved January 4, 2023, from https://github.com/Ileriayo/markdown-badges

[8] The Java™ Tutorials: Coordinates. Oracle. Retrieved December 10, 2023, from https://docs.oracle.com/javase/tutorial/2d/overview/coordinate.html