INTERNATIONAL UNIVERSITY

VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY

School of Computer Science and Engineering

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

PROJECT REPORT

CANDY ESCAPE

Algorithms & Data Structures (IT013IU)

Semester 1 - Academic year 2023-2024

Course by Dr. Tran Thanh Tung

TABLE OF CONTENTS

CONTRIBUTION TABLE

No.	Full Name	Student's ID	Contribution
1	Hoàng Gia Huy	ITCSIU21186	100%

ABSTRACT

A match-3 game is a captivating puzzle genre where players strategically swap adjacent tiles or objects to create a row or column of three or more identical items, causing them to disappear and new ones to fall into place.

Originating in the early 2000s, with titles like "Bejeweled" setting the trend, match-3 games have since evolved into a diverse and popular genre on various platforms. The gameplay is simple yet addictive, requiring players to think ahead and plan moves to achieve higher scores or complete objectives. Whether it's gems, candies, or other themed elements, the goal remains the same – match, clear, and progress through levels.

Chapter 1: INTRODUCTION

A. Objectives

The project aims to recreate a fully playable 2D match-3 game. This game can also demonstrate core structure for the Algorithms & Data Structure (DSA) course.

In short, this project aims to:

- Create a redesigned game to entertain.
- Apply all the data structe in DSA course theory in real life.
- Have a brief look at game development, code optimization, and project management.
- Learn to build and manage a game project.
- Evaluate the ability to build more features on the base program.

B. The tools and programing language used.

- IDEs for programming and debugging are Unity 2022.3.15f1 version and Microsoft Visual Studio. (Figure 1.1 and 1.2)
- Code version management : GitHub (Figure 1.3 and 1.4)
- Project management: Github (Figure 1.2)
- Programing language: C#

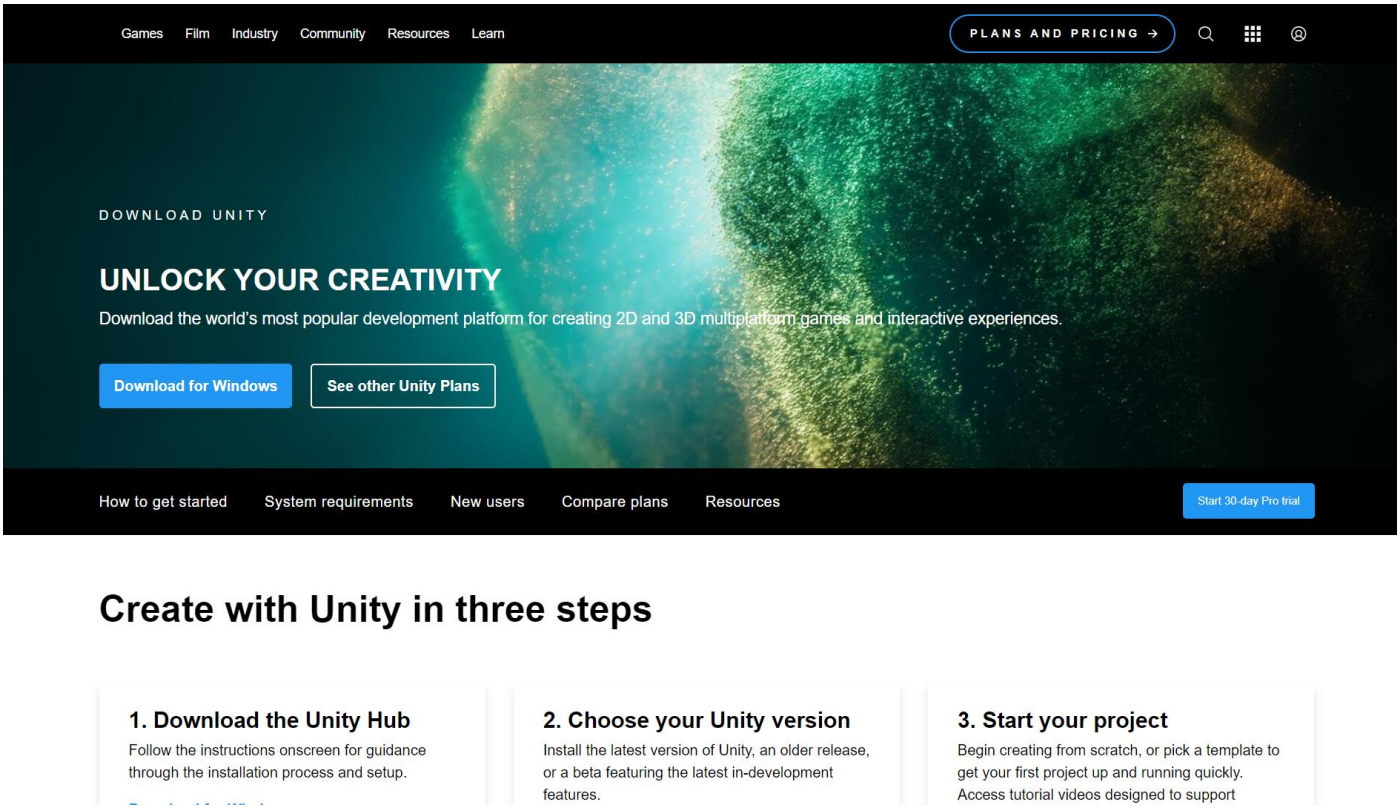

Figure 1.1. Unity Download Page

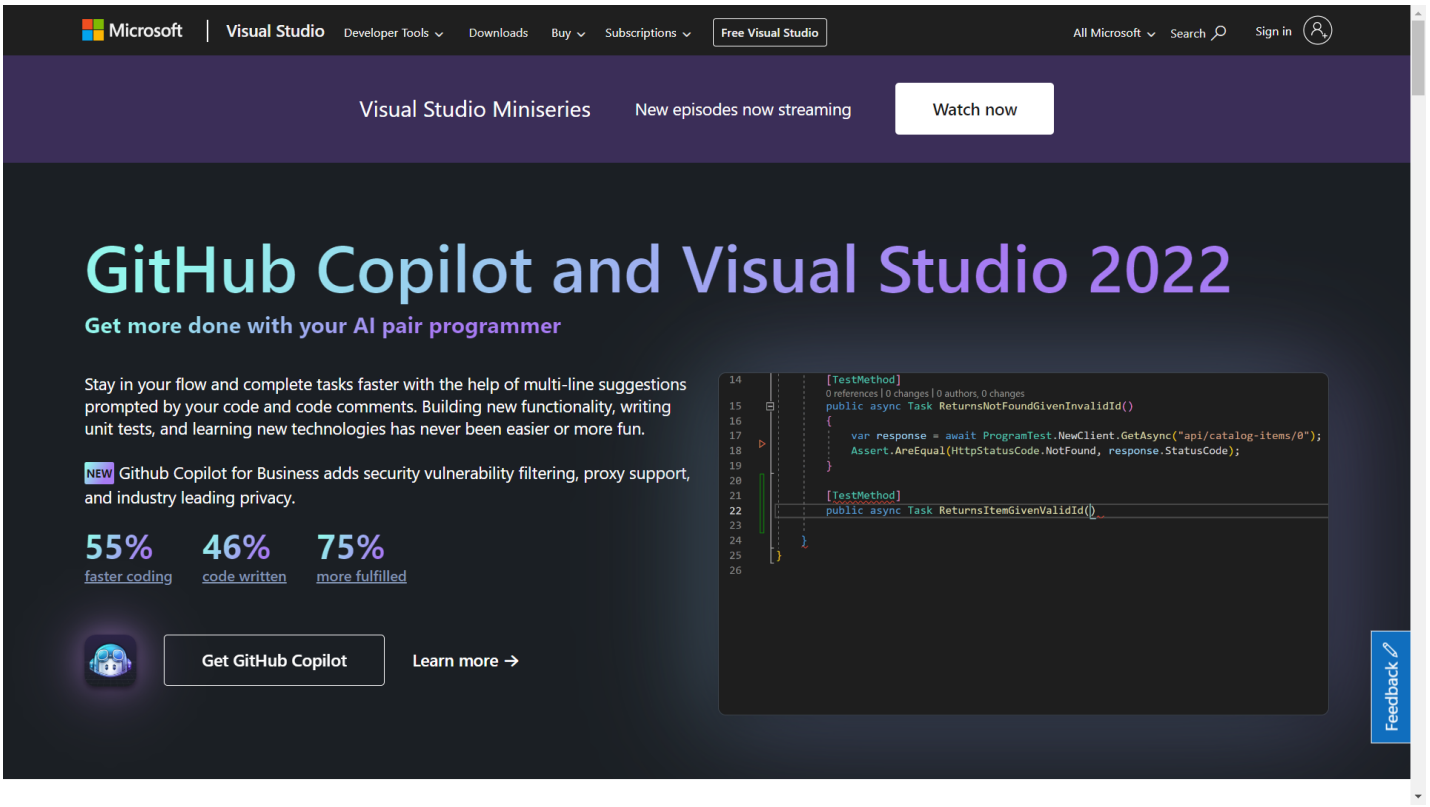

Figure 1.2. Microsoft Visual Studio

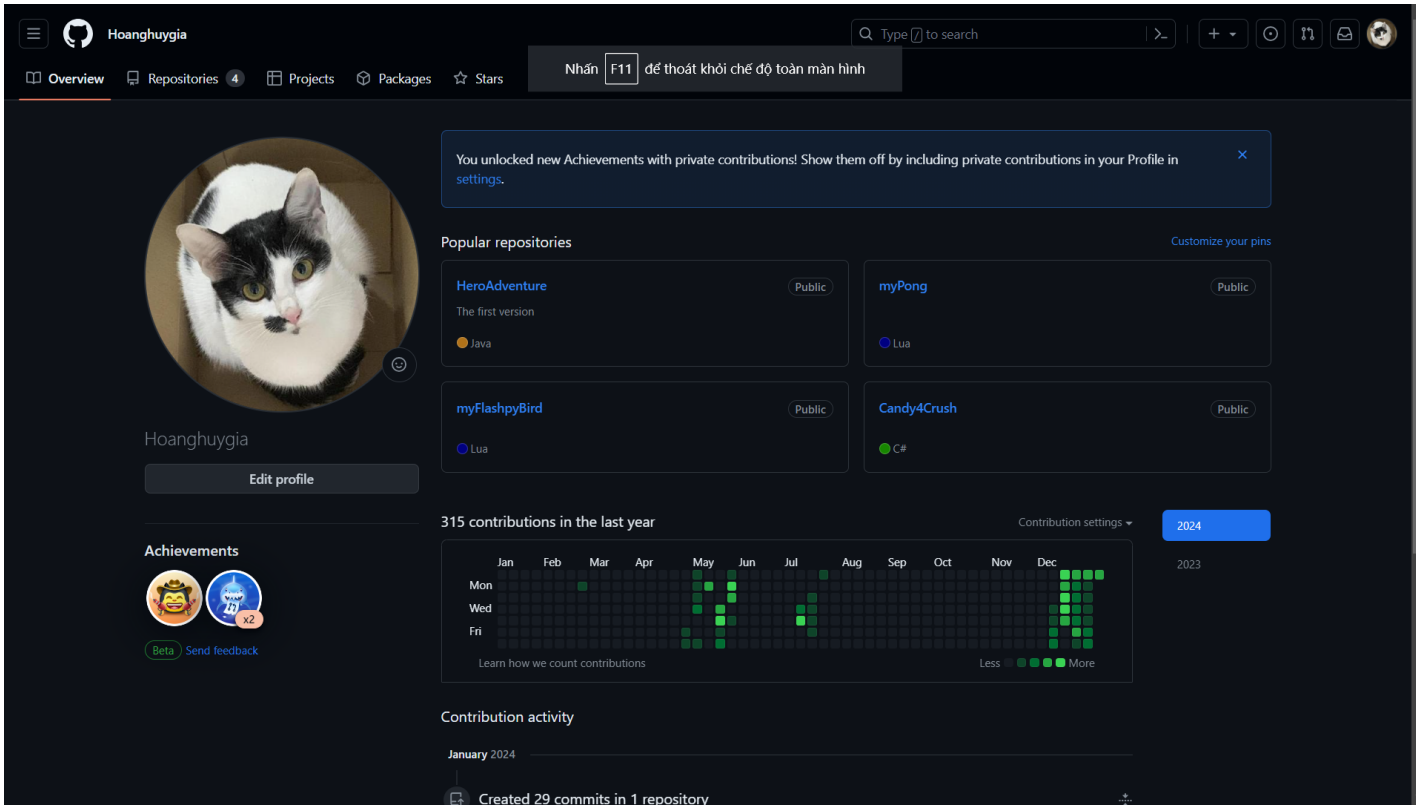

Figure 1.3 My Github Profile

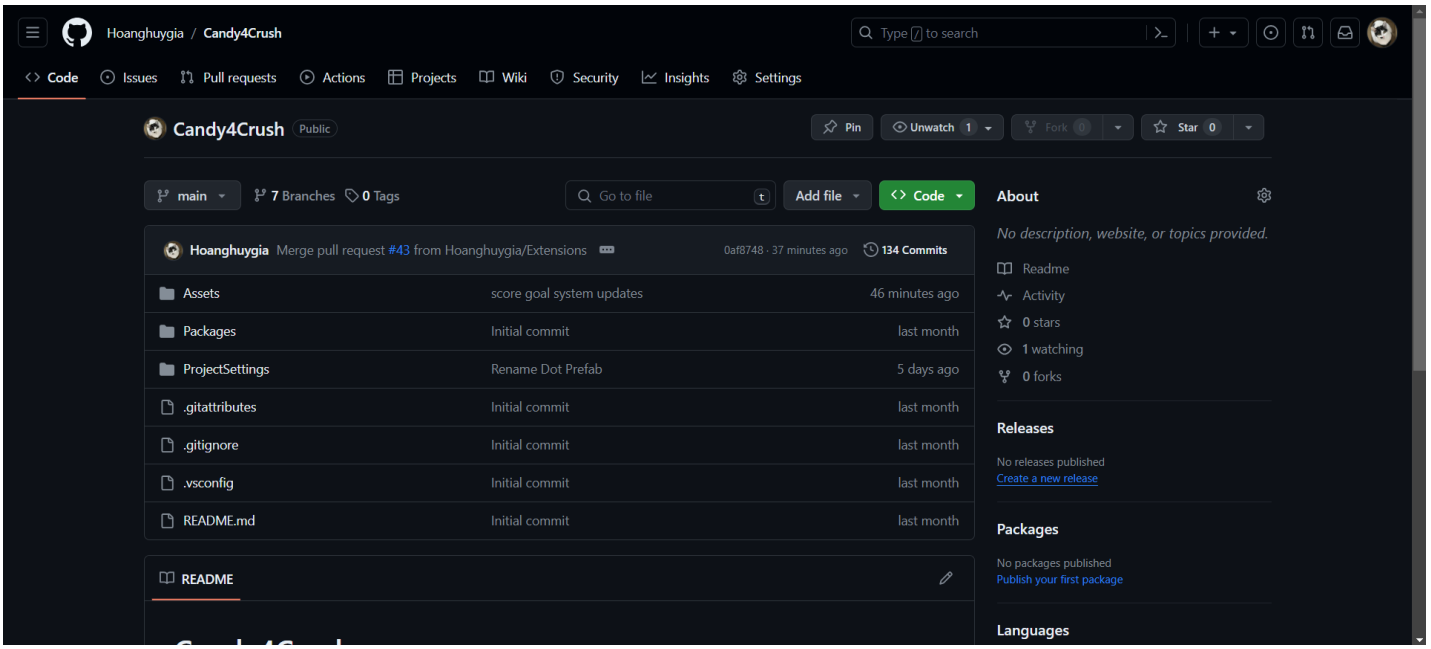

Figure 1.4. Github Project

Figure 1.5 Project contribution

Chapter 2: METHODOLOGY

A. Gameplay

The **gameplay** is straightforward: You (the player) goal is to destroy all possible matched pieces on the board of the game. Use the finger to swipe the pieces. If the matched pieces is more than three, it would be destroyed from the board and random color pieces would be loaded. This game has 3 levels (due to the lack of time, I only develop three levels to demo), each level has individual level goal that player must reach to win. Also, each level also has score goal to get stars, the maximum star for each level is three. You can accumulate it to show your ability. Moreover, the game also have special bomb that could destroy larger than three pieces at a time. This is:

+ **Row bomb**: this kind of bomb destroy all the pieces of a certain row.

+ **Column bomb**: this kind of bomb destroy all the pieces of a certain column.

+ **Adjacent bomb**: This kind of bomb could destroy all the next pieces of the pieces as considered as adjacent bomb. The maximum destroyed is nine pieces.

+ **Color bomb**: This kind of bomb is to destroy all the bomb that has the color of the pieces that we swipe into.

E.g When we swipe the color bomb to the red pieces. This would destroy all the red pieces on the board.

After considered the basic rule, we come to play. The first screen is the main screen (Figure 2.1). On this screen we have two button, the first is the Tap To Play button at the center of the screen. Click this button, we could enter to the level select screen. The second button of this creen is the setting button (the blue one, on the right top of the screen), when you click this button, this would allow you to control the sound/music stuff (Figure 2.2). Moreover, you all could see the name of the creator

(in this case is me) . Other two option is to login by gmail and vibrate mode, however, these function I still not developed yet.

The next screen is select level screen (Figure 2.3). When you come into this screen, there are totally 9 levels for you to choose, however, I just developed three levels. In the future, maybe this would have more level, with different difficulty. Besides, nine level as basic screen I set, we have 3 more buttons, the main button (blue one - on the center below is to back to prevous screen). The other two button is button to the next page and the prevous page of the level select screen. However, this two button only possible when we have more than nine levels!.

After choose the level you one (for the first time that you play, you could only choose the level 1, you must pass it before choose next level). This would leak you to the confrim panel – where you could see the **maximum** star and point of this account (Figure 2.4).

This panel has two button, one for back to prevous screen, one for confirm play the level (Figure 2.5). Moreover, this panel also show the target, the color candy girls that you need to collect.

The last one is the play zone, after confirming playing, you would come to the play ground (Figure 2.6). Where has a board with different candy (in this case is different candy girls with different color). Your goal is to destroy the goal of the level. If you collect enough candy girls as the targe, this would leak you to the win screen (Figure 2.7) – where you could back to the main screen. This action also open the new level for you to finish!. However, as other game, this game also have the condition to win, there are two modes. The first one is Moves mode, this would limit the moves that you can have. The other modes is time mode, you only have a certain time to collect!. Moreover, this scrren also provide a way to go out, this is setting panel on the right

top of the screen, (the blue read one) (Figure 2.8) and move you to the lose screen (Figure 2.9)

. Click it, you would have three options, turn on/ turn off sound and music, out the game, …

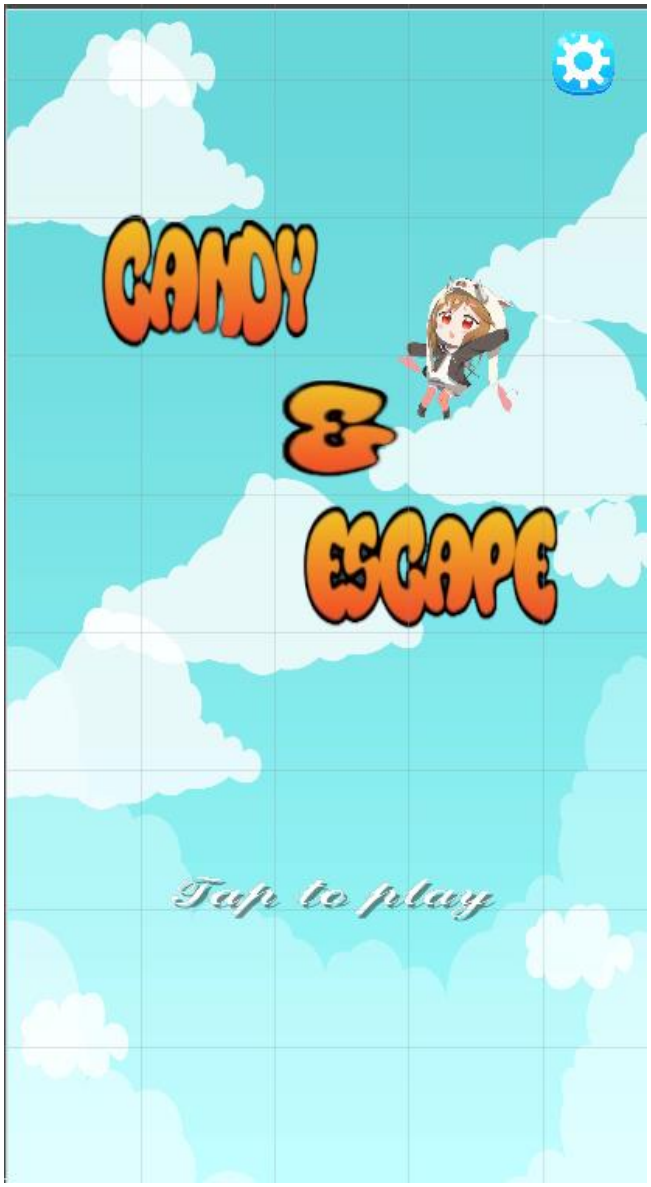

Figure 2.1. The main screen

Figure 2.2. Inside Setting button

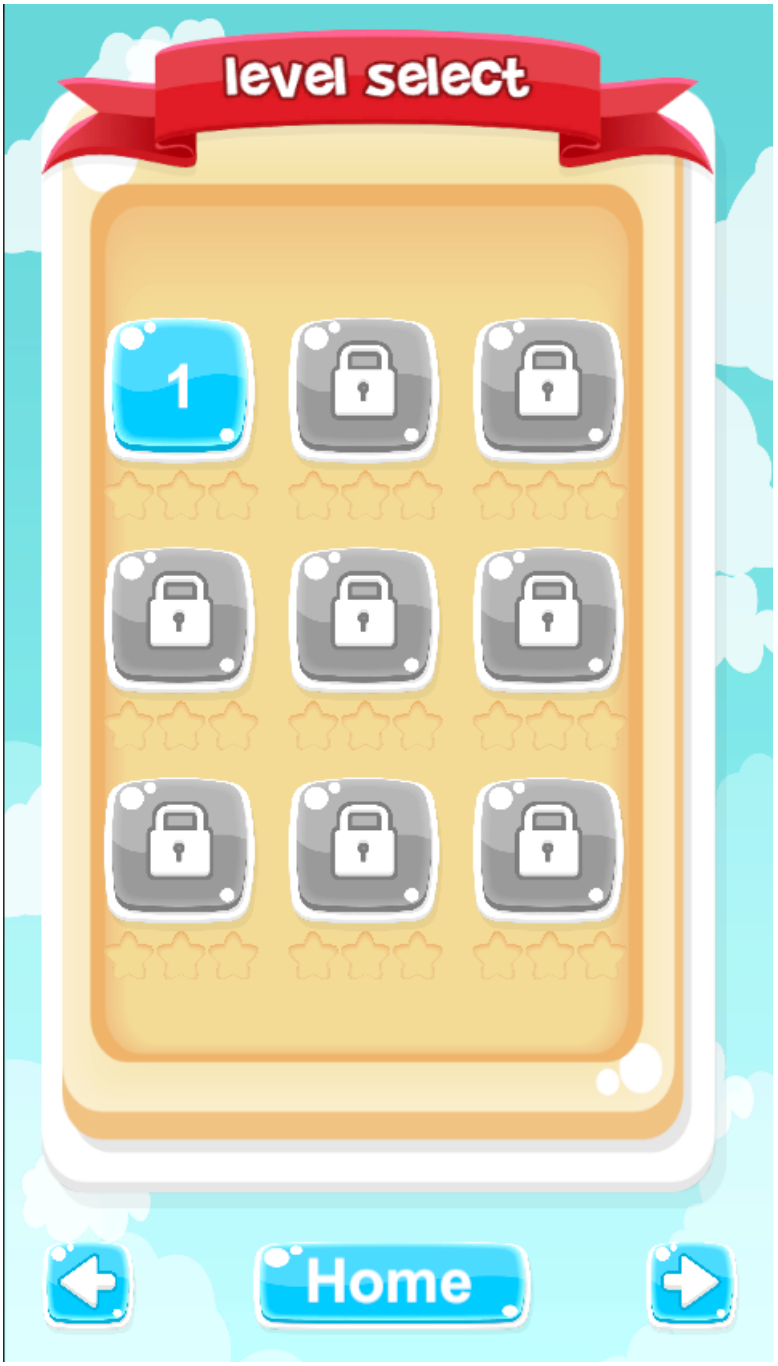

Figure 2.3. Level select panel

Figure 2.4:The confirm panel.

Figure 2.5. Goal view for the level

Figure 2.6. The play ground

Figure 2.7. Win panel

Figure 2.8. Inside te setting button, the red one is to out current game

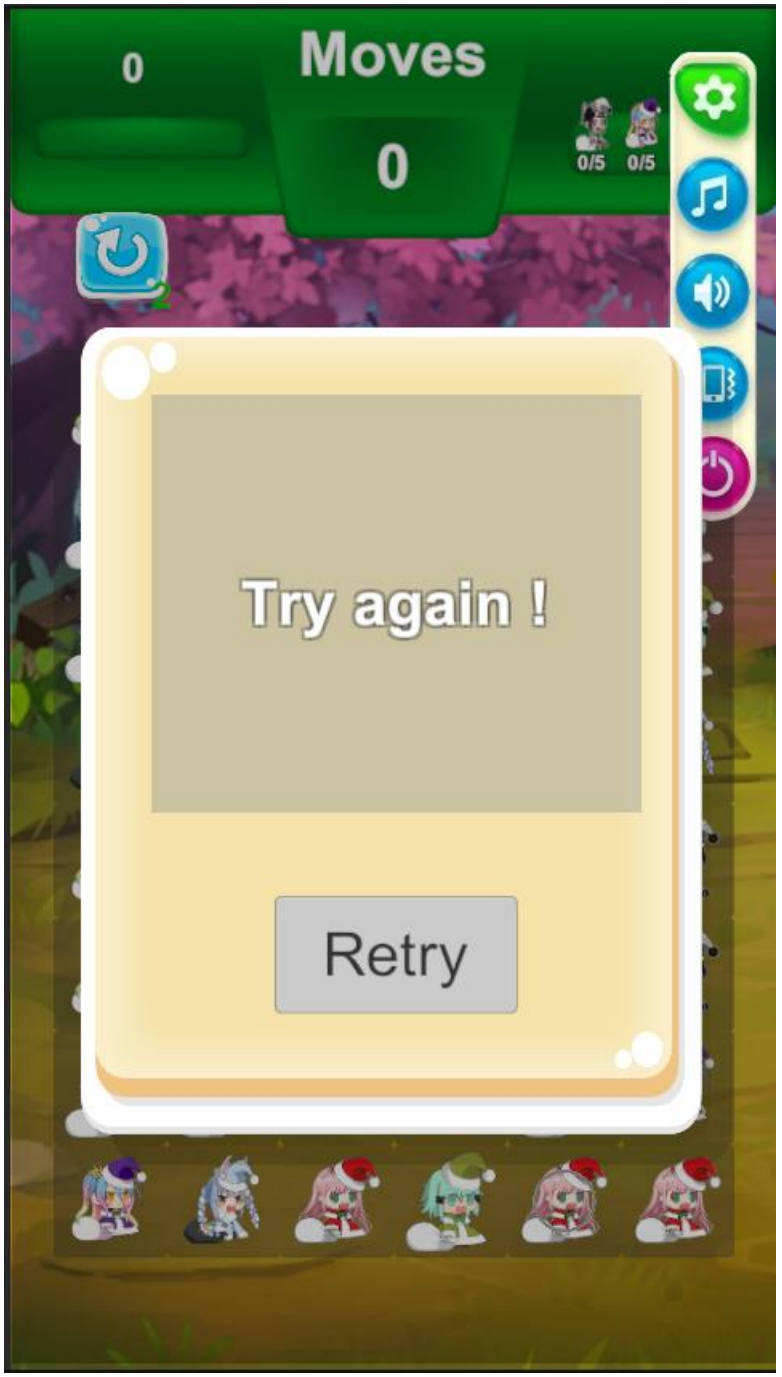

Figure 2.9 The lose screen

B. Design

- In this game, mostly the candy girls is the image that I find it interesting and decided to use it in my game (Figure 2.7).
- Other stuff like row bomb, column bomb, adjacent bomb, color bomb only image that I randomly serach on the internet and not change to be suitable to the candy girls.
- Besides the pieces, the UI is much more difficult for me, I collect it mostly in the freen UI page that I give in the reference (Figure 2.8 & 2.9).
- To make animation for movement or explosion, we had to draw frames by frames and put them in a loop.

Figure 2.10. The princesses art.

Figure 2.11. The top UI

Figure 2.12. Some window panel UI

C. UML Diagram

The most important class in this project is Dot.cs and Board.cs. The first one is to define the dot or pieces. It includes all the attribute, also the behavior of the dot from the way it swipe or the destroy animation. In the other hand, the Board.cs class is to define the behavior of the board or the skeletion of the game (Figure 2.12 & Figure 2.13). It defines and controll all the stuff such as sound, start panel, win panel,…. All the other class and fuction are revoke in this class by some other independent class . It is also in charge of updating and drawing the status of each object such as pieces, panel, …. This class is basically the core of our project.

Moreover, there still some crucial classes such as FindMaches.cs (Figure 2.14), this class we could managely see it as the expandsion of the dot class. This class focus on the how the dot interact with each other, and the way it behaves.

The SettingButtonController.cs and FadePaneController is to focus on the button system in the main screen (Figure 2.15). They are the behind of how the button in the panel above the board interact with the player, how it moves from panel to panel.

Other important class could be the TileScreeenPanel.cs and LevelButton.cs, this is the same as the two classes above but it control the button of the main screen and level select screen.

Another important class is EndGameRequirement.cs, this class is to decide the behave of the game when it meet the requirements such as win game or fail the level.

Finally, these classes help us to paint the animation, check position, and some

keyboard events.

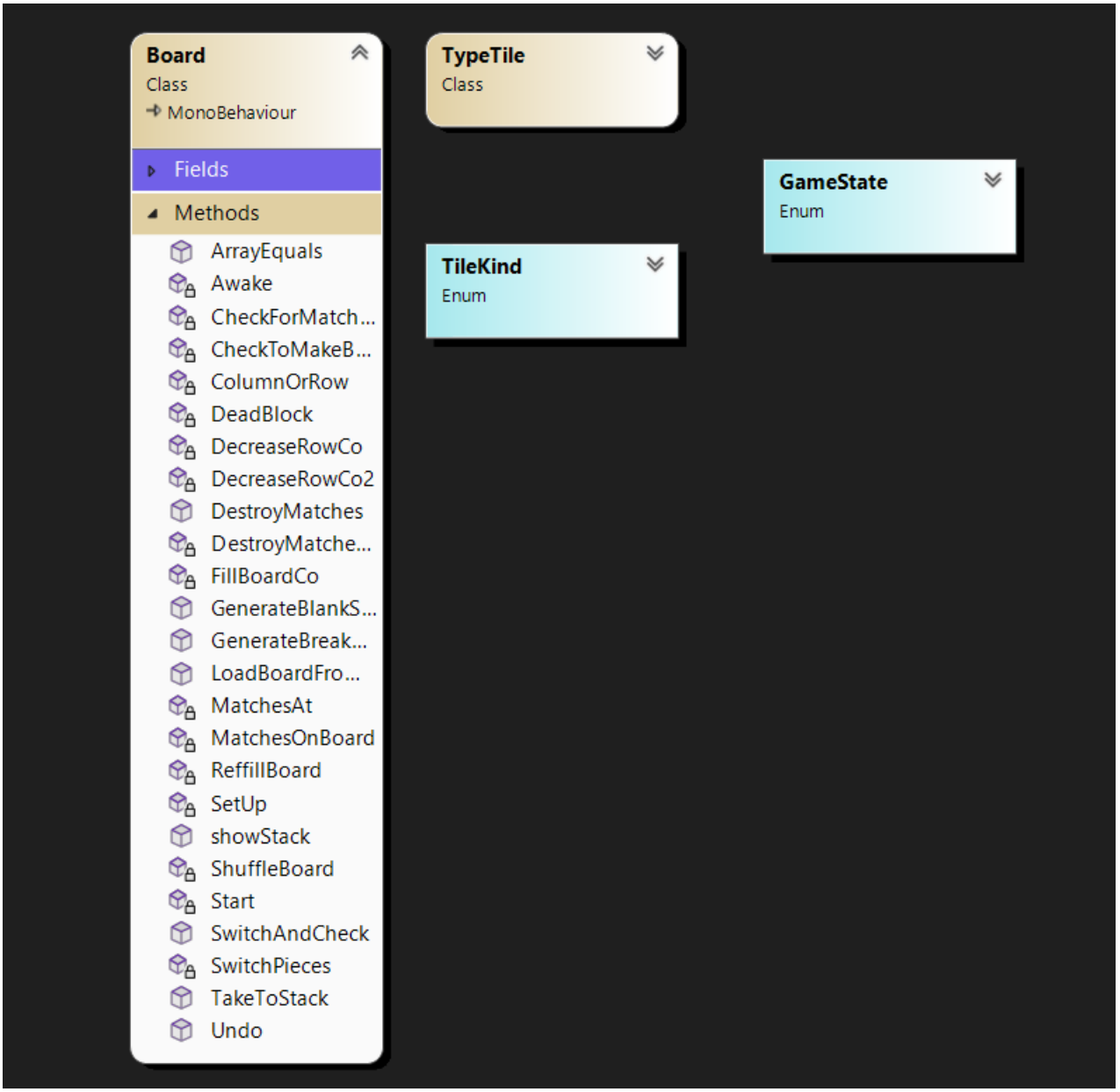

Figure 2.12. Board class and its subclass

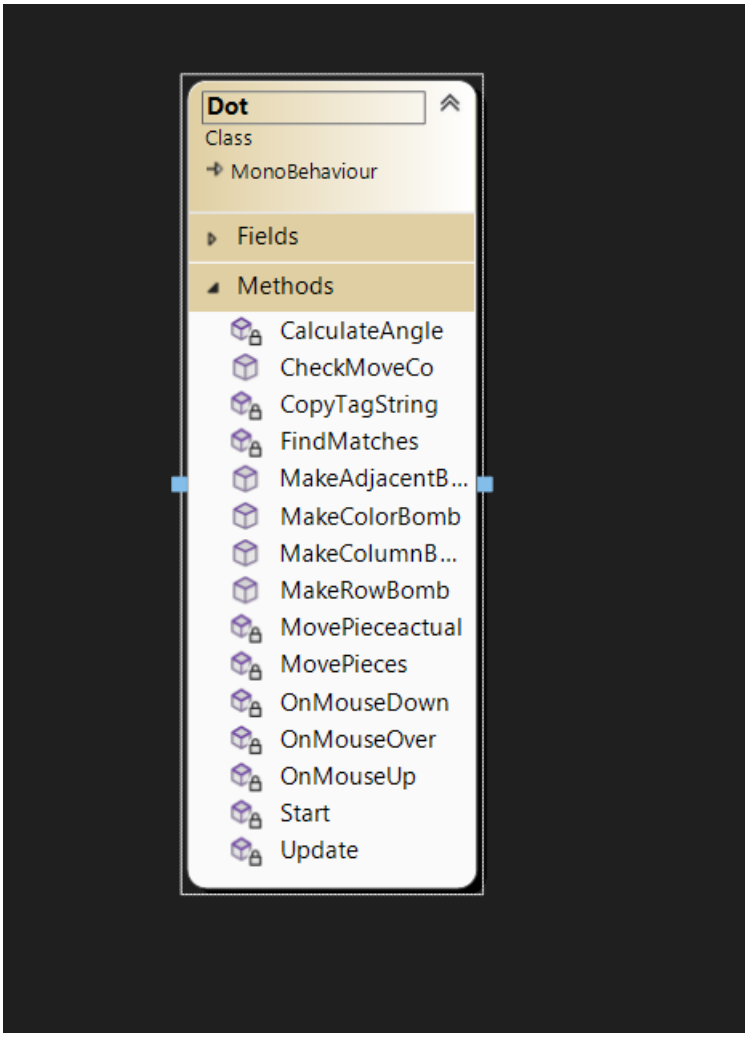

Figure 2.14. The dot class

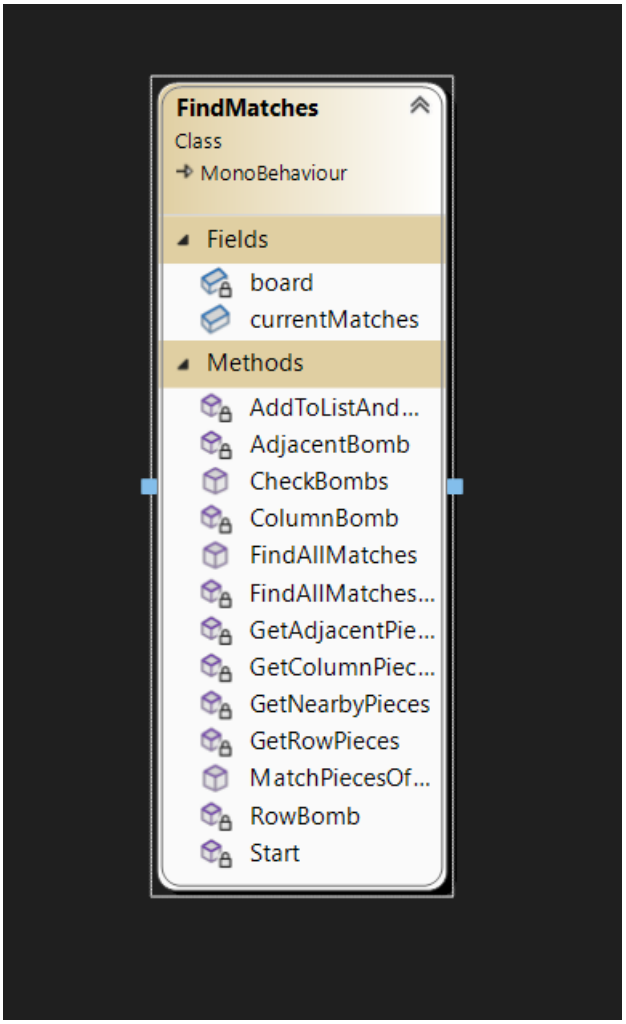

Figure 2.14. The FindMatches class

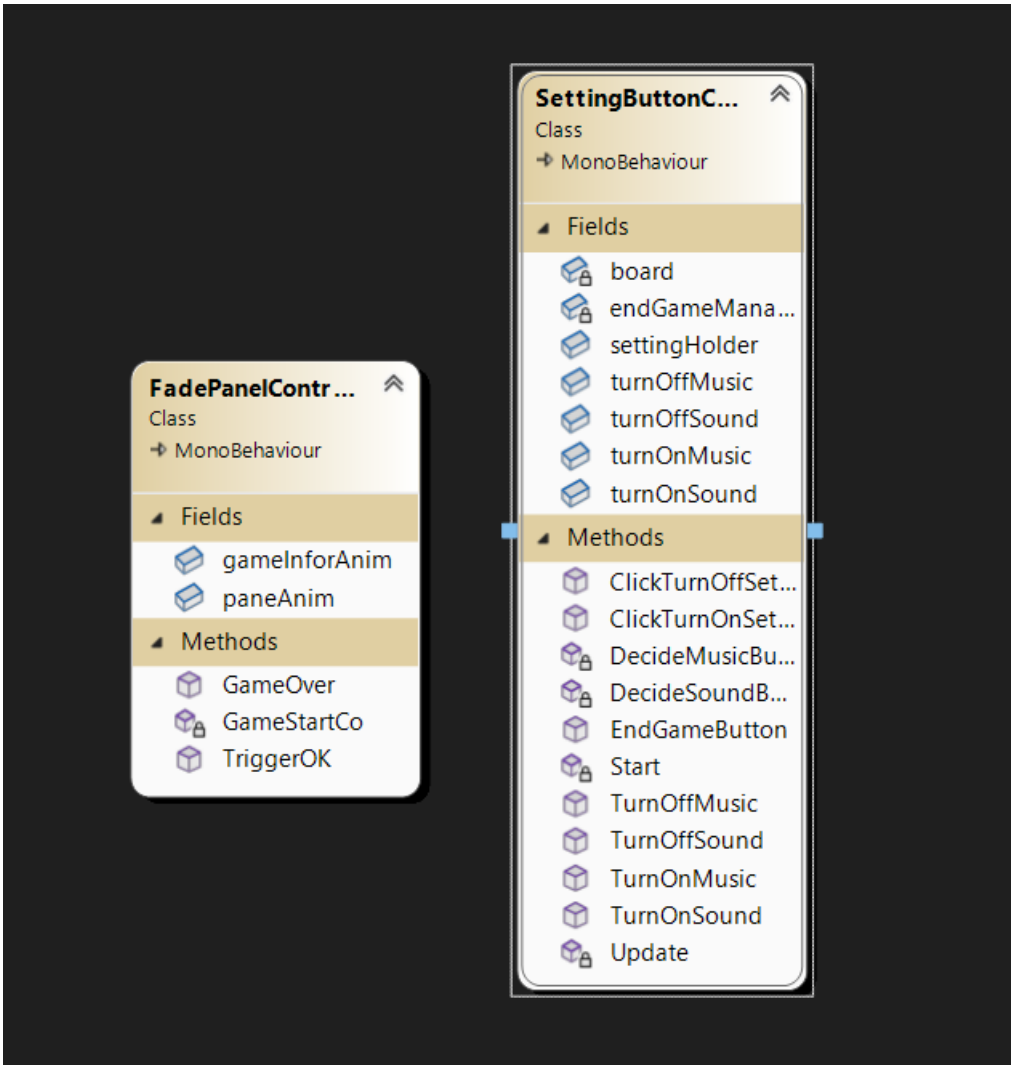

Figure 2.15. SettingPanelController and SettingButtonController class

D. Data structure and Algorithm

In this game, we use two main thing from the DSA course is array and stack.

For array, we use most for store the board. For example, I used 2D array to store the board, with two dimension is the width and the height of the board. Moreover, I also use array for much more features such as score, level list, ….. These array a crucial part of the game, without it, we could not manage the game, especially in unity – which array could be manage in Unity. Besides, we could use Linked List instead, however, it would not very suitable in Unity and take longer time since most of the work we do for array is to recall not search, insert,…

Other thing we use is stack structre (Figure 2.16). I use stack for the Undo feature, since

stack is First Come Last Out, so we could store the first swipe at the deepest place. It means that we coud easily recall the lastest change.

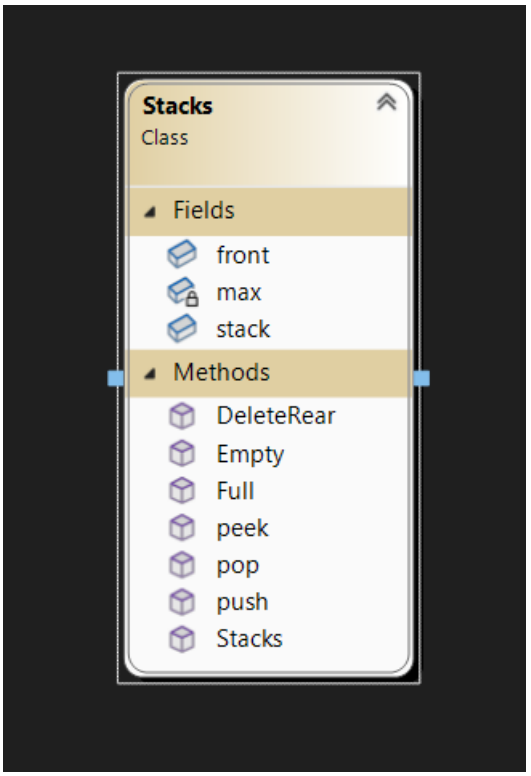

Figure 2.16. The stack class

Chapter 3: CONCLUSION

A. Summary

In conclusion, the game project has all four features of DSA course as well as the OOP feature, the SOLID principles, and some design patterns, such as Command.. Due to the limited acknowledgment time, members must simultaneously learn and implement new things in the code. The code still needs to be optimized for fluency. However, it was a very memorable experience for our team to learn new knowledge, practice, discover how to work as a team, manage a project, etc.

B. Shortcoming

Due to the limited development time and knowledge, the game could have been more optimized, and all of our programs are still running on one thread. Moreover, there some things that could be improved such as the algorithm to destroy the pieces. It stll has some bugs inside – which I need more time to explore and fix. The UI is not realy friendly to user, for example, it do not satify the player. There some cracks insides,…

C. Future Works

We only have some limited time for the project; therefore, many features still need to be implemented. In the future, we want to upgrade the game with more

useable items like potions to increase time, moves and increase temporary player's damage to breakble tile. We also want our pieces destroy animation to be better. We also want to design a system to upgrade a list of item that we could use for the game such Timer, Portion,….

Furthermore, we did not make the most use of SOLID Principles and various Design Patterns to improve the performance of our code, which makes code very hard to make extensions.

REFERENCES

https://www.w3schools.com/csharp/

https://www.shutterstock.com/image-vector/

https://www.watermarkremover.io/upload

https://opengameart.org/content/free-game-gui

https://www.youtube.com/watch?v=FnOR--bKdi8

https://youtube.com/playlist?list=PL4vbr3u7UKW

xEz75MqmTDd899cYAvQ_B&si=CAYIFAxxD4

FFB