GAME PROJECT REPORT

**2048 GAME**

**Submitted by**

Nguyễn Thế Duy Khanh ITITRG17016

Võ Nguyên Khôi ITITRG17008

Hà Khánh Lâm ITITWE17003

Bùi Hoàng Thiên Bảo ITITWE17026

Hồ Thái An ITITRG17017

Date Submitted: [4/6/2019]

Course Instructor: [Tran Thanh Tung]

Table of Contents

List of Figures 2

List of Tables 2

Introduction 3

List of contributions…………………………………………………………………….. .4

Project management…………………………………………………………………… …...4

Game structures………………………………………………………………………… .5

Game design 13

References 19

List of Figures

Figure 2: Contribution graph 3

Figure 4.1.1: Menu Structure 5

Figure 4.1.2: Button Structure 6

Figure 4.1.3: Introduction Structure 6

Figure 4.1.4: Option Structure 7

Figure 4.2.1: Initialize the board 7

Figure 4.2.2: Creating a random-value block 8

Figure 4.2.3: Game loop function 8

Figure 4.2.4: Moving function 9

Figure 4.2.5: Merge the blocks 10

Figure 4.2.6: Score and high score 10

Figure 4.2.7: Undo structure 11

Figure 4.3.1: Check winning structure 11

Figure 4.3.2: Losing conditions 12

Figure 4.3.3: Win screen and lose screen structure 13

Game design

Figure 5.1: Introduction of menu 14

Figure 5.1.1: Game script . 14

Figure 5.1.2: Sound setting 15

Figure 5.2: Game interface 15

Figure 5.3.1: Win screen 17

Figure 5.3.2: Game over screen 17

List of Tables

Table 1 4

Table 2 5

1. Introduction:

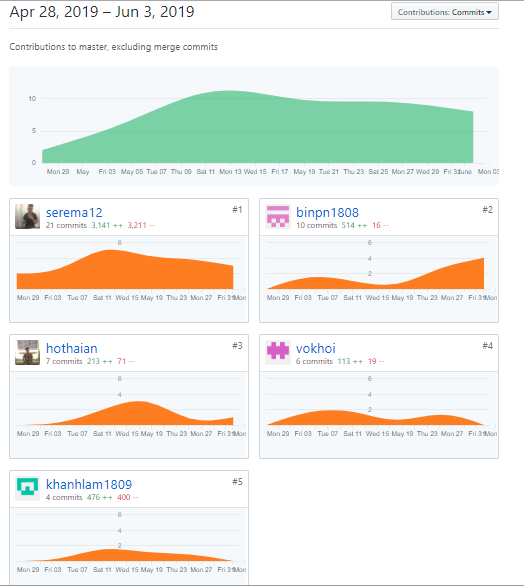
**2048** is a single-player [sliding block puzzle](https://en.wikipedia.org/wiki/Sliding_block_puzzle) game designed by Italian web developer Gabriele Cirulli.

The game's objective is to slide numbered tiles on a grid to combine them to create a tile with the number 2048. However, one can continue to play the game after reaching the goal, creating tiles with larger numbers.

1. List of contributions:

|  |  |  |
| --- | --- | --- |
|  | **Member** | **Assignment** |
| **1** | Khanh | Make the game basic (move, draw board, …) |
| **2** | Khôi | Create a menu for the game, make report. |
| **3** | Bảo | Make the game interface, PowerPoint and restart function |
| **4** | Lâm | Make PowerPoint, Undo function and calculate the score |
| **5** | An | Make report, end game screen, store highest score |

Table 1



**Figure 2**: Contribution graph

1. Project management:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Works** | **Done by** | **Deadlines** |
| **1** | Menu | Khôi | 6/5/2019-19/5/2019 |
| **2** | Game Interface | Bảo | 6/5/2019-19/5/2019 |
| **3** | Calculating score | Lâm | 6/5/2019-19/5/2019 |
| **4** | Restart function | Bảo | 6/5/2019-19/5/2019 |
| **5** | Undo Function | Lâm | 6/5/2019-19/5/2019 |
| **6** | Game, Win screen | An | 6/5/2019-19/5/2019 |
| **7** | Highest score | An | 6/5/2019-19/5/2019 |
| **8** | Report | Khôi, An | 25/5/2019-29/5/2019 |
| **9** | PowerPoint | Bảo, Lâm | 25/5/2019-29/5/2019 |

Table 2

1. Game Structure.

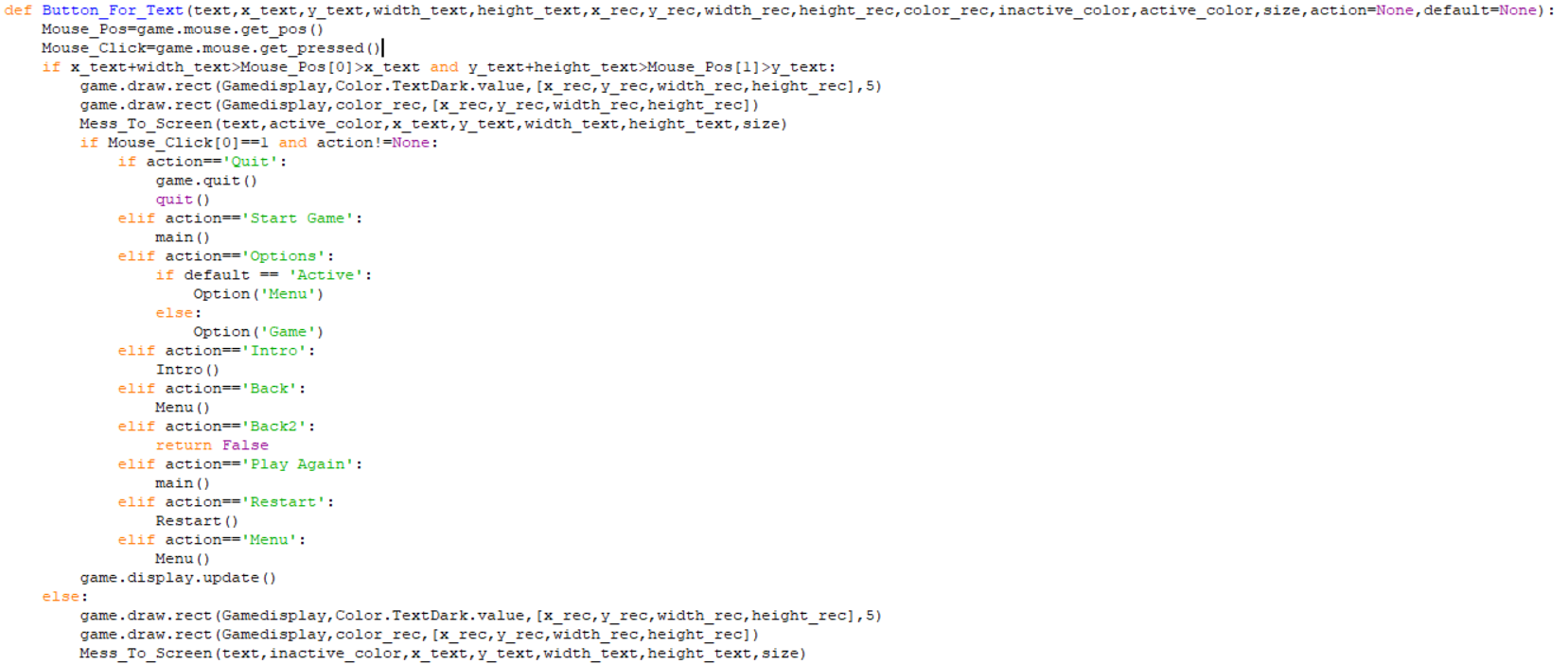
We created the game with the help of python 2.7.14 , pygame library and Cx\_Freeze.

1. Menu Screen

The Button\_For\_Text function is used to create buttons for the menu and each button will trigger a different function of the game. 

**Figure 4.1.1:** Menu Structure

Each button will be compared with the action by the function “if” within the Button\_For\_Text function to go to the action that we created.



**Figure 4.1.2**: Button Structure

1. Introduction

Introduction part displaces the rules of 2048 game.

* Moving box: Using up, down, left, right arrows.
* Undo actions: press “u”.
* Restart action: click to “restart” button.

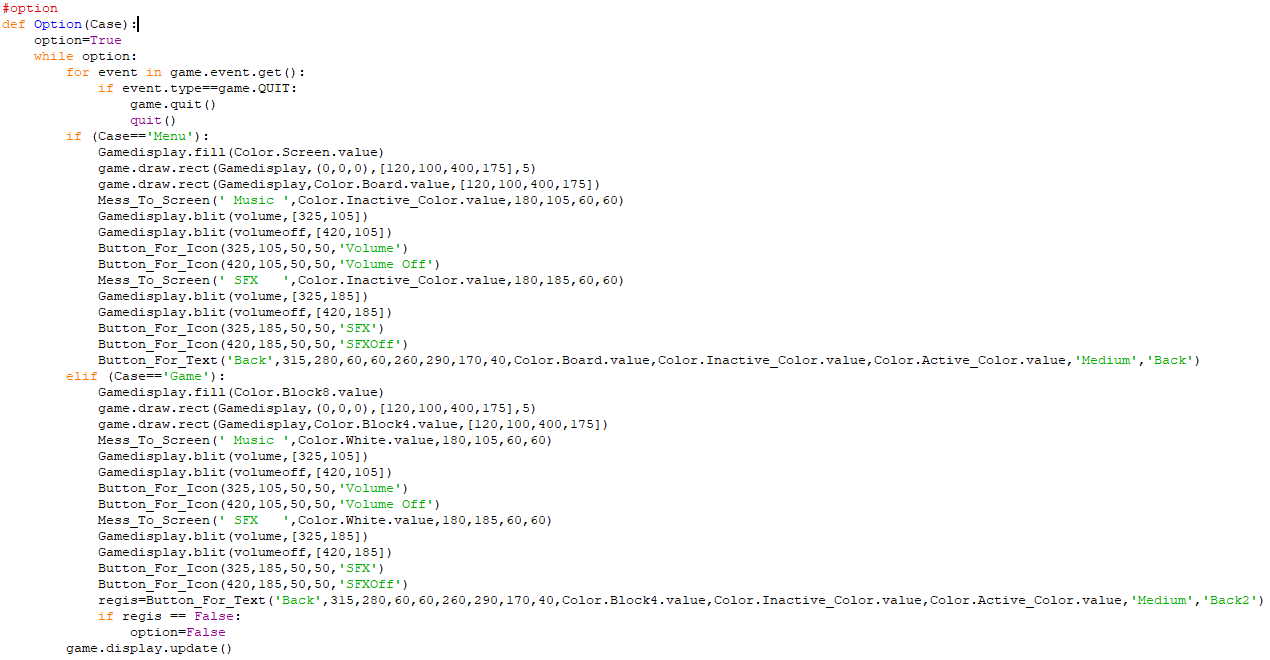


**Figure 4.1.3:** Introduction Structure

1. Option

The Options devides into 2 parts due to the fact that the game includes 2 other interfaces for the Option in menu and in the main gameplay.

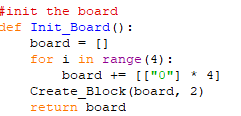
The main function of both option menu is to adjust the game’s music and sfx.



**Figure 4.1.4:** Option Structure

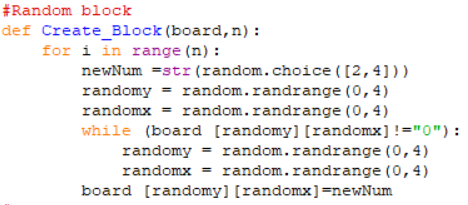
1. Game play
2. Initialize the board

The function creates 2 dimensional array which is filled it with ‘0’ value (empty blocks) and create 2 initial blocks with the value of 2 or 4.



**Figure 4.2.1:** Initialize the board

Create block to initialize the number of blocks with the value of 2 or 4.

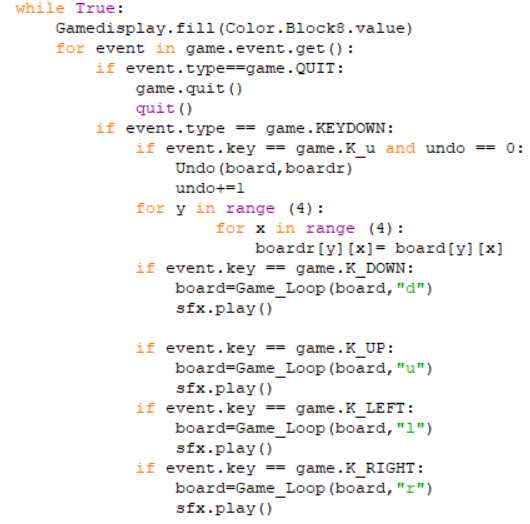
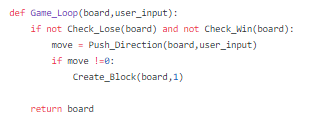


**Figure 4.2.2:** Create a random-value block

1. Gameplay

The system of the game will be executed in the following orders:

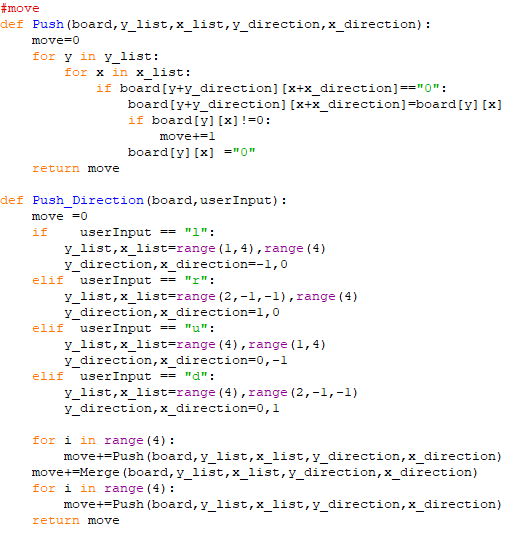
* Each time the game receives the input of the user, the game will check if the game reaches the win or lose condition, if not then it will initialize a block with the value of 2 or 4.



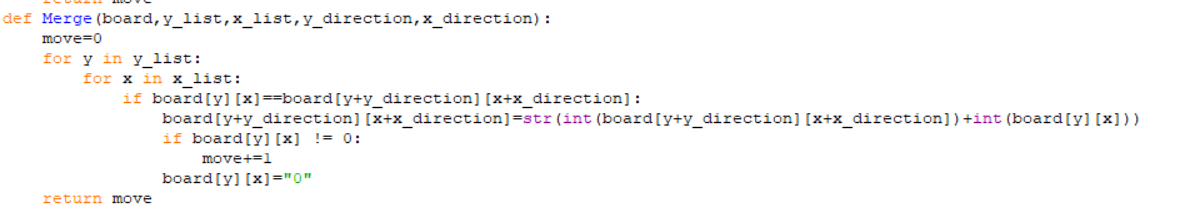
**Figure 4.2.3:** Game loop function

* By pressing the arrow keys, the game will automatically move and be

compared with the input arrows within the conditions in Push\_Direction function and Merge if the next two blocks in the input direction are equal to each other.



**Figure 4.2.4:** Moving function

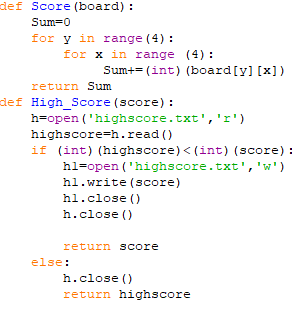


**Figure 4.2.5:** Merge the blocks

1. Score and High Score

The Score function will sum up every element from the current board and add

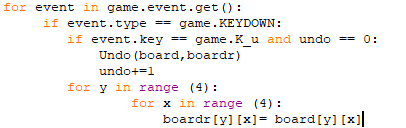
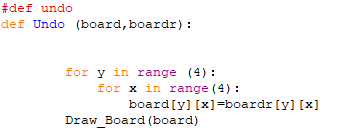
it to the ‘Sum’ variable then compared it to the highest score value which is stored in a txt file. If it is higher than the highest value, then the score will be overwrote with the original value in the txt file .



**Figure 4.2.6:** Score and high score

1. Features

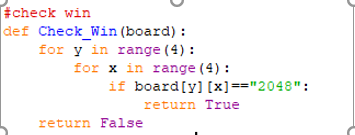
To undo , we construct a temporary board to store the current board in ‘board’ before making the next move. The key “u” is declared the key to trigger the undo function.



**Figure 4.2.7:** Undo feature

1. Win Lose
2. Winning condition

To check if the player wins, we check if any value in the 2-dimension array is equal to 2048.

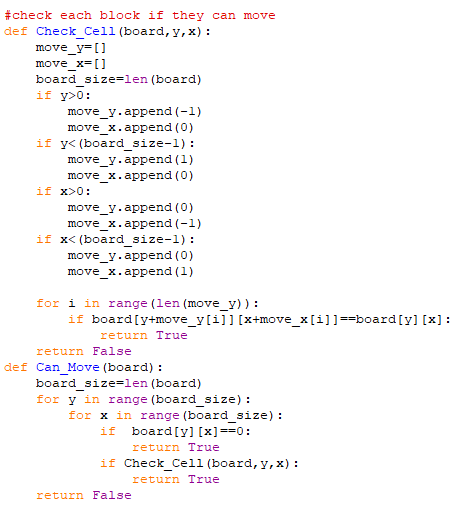


**Figure 4.3.1:** Check winning structure

1. Losing condition

To check if the player loses, we check:

* Each block can be move up, down, right, left (check if the side-by-side block in fourth direction are equal to the current block that we are checking, if true than we can still move). (Check\_Cell function)
* The value of each block is not a no-value block

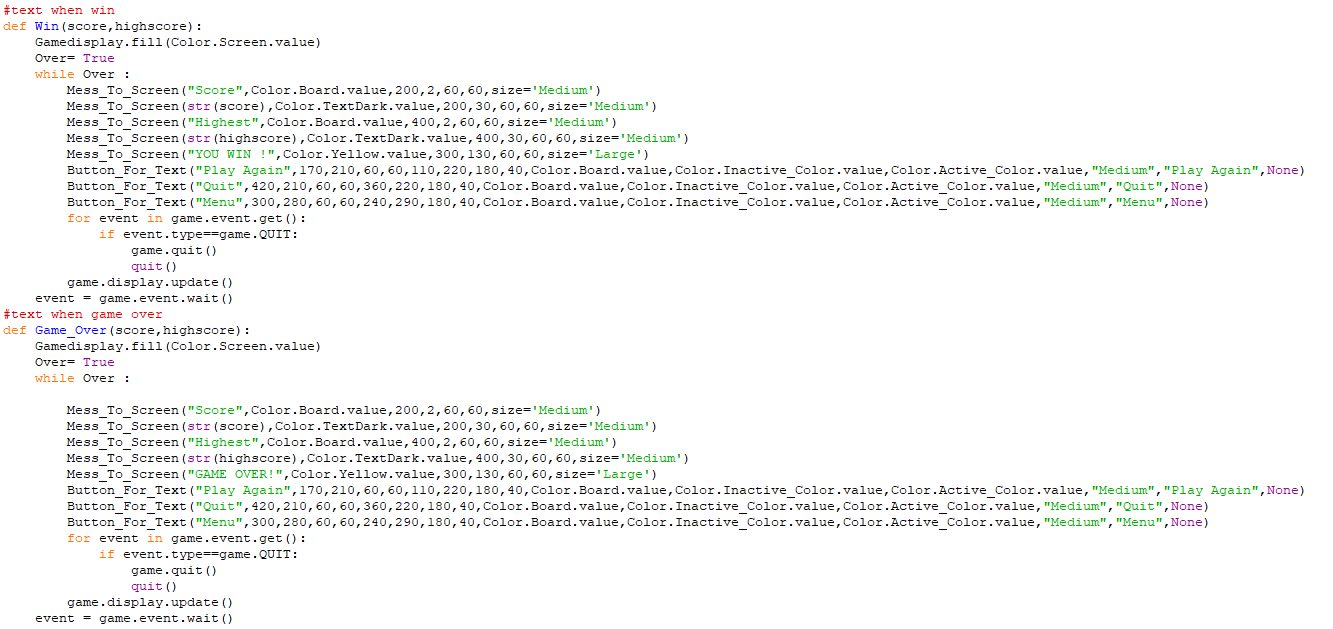


**Figure 4.3.2:** Losing conditions

1. Win and lose screen

If the condition win or lose reach, the game will be change into the Win Screen

or Lose Screen respectively.



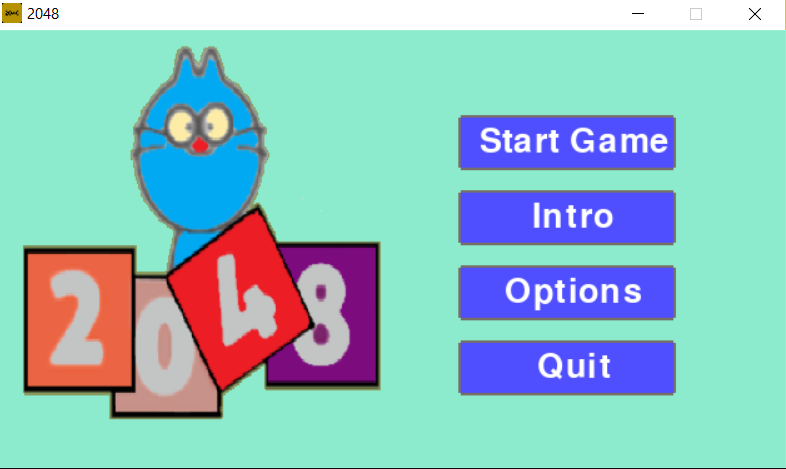
**Figure 4.3.3:** Win screen and lose screen structure

1. Game Design

Our game includes 3 main screens: Menu Screen, Main game screen and Win (Lose) Screen.

1. Menu Screen

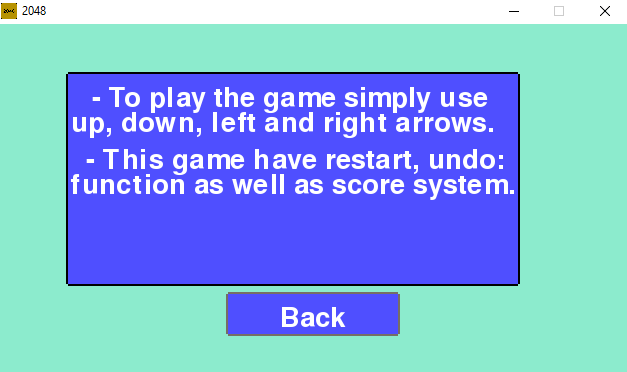
The menu screen includes 4 clickable buttons for different function of the game with some simple decoration.



**Figure 5.1 :** Introduction of menu

1. Introduction

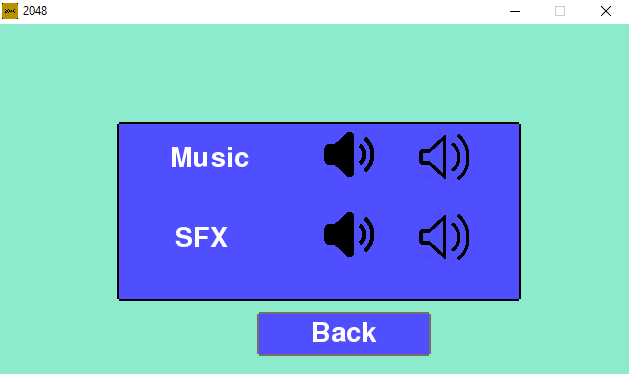
The Intro button is a short introduction about the game with how to play and some function that are included in the game.



**Figure 5.1.1:** Game script

1. Options

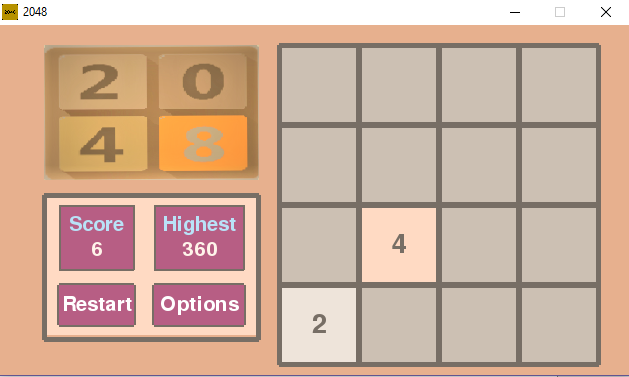
The Options button is used to control the music volume and sfx volume of the game.



**Figure 5.1.2:** Sound setting

1. Game Screen

To start the game, click the start game button which is the first button from above. After clicking the button, the screen will change to Main Game Screen



**Figure 5.2:** Game interface

The Main Game Screen includes a main game board where the actual game is play. The aim is to merge the increasing block from 2 to 2048 and reach 2048 to win the game.

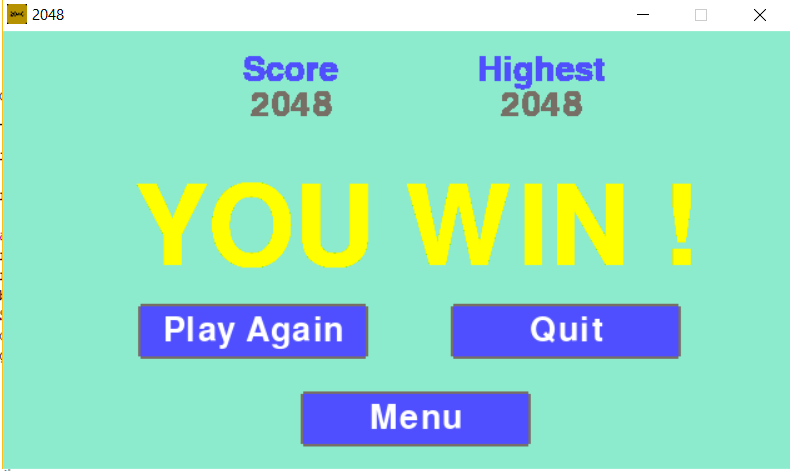
On the left side is the score which is the player is having and the highest score is the score which is the player has to overcome. Moreover, the restart function‘s purpose is to restart the game and the options as mentioned above ,to turn on and turn off the SFX and Volume.

While playing the 2048 Game, press “u “ on the keyboard to use the Undo feature function .

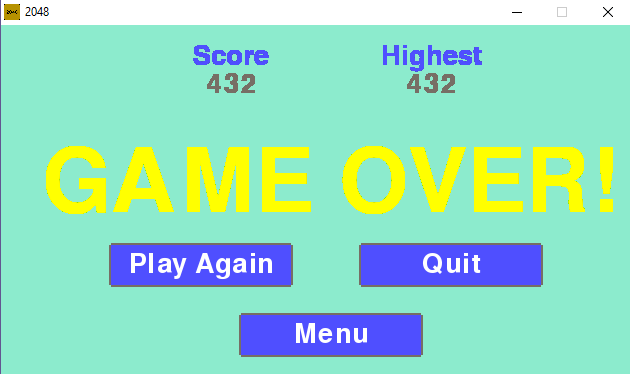
1. Win Lose

The Win (Lose) Screen will display the result of the player compared to the highest score and show the result of the player in the both circumstances: Win or Lose.

Moreover, by pressing play again, it will restart again or by pressing quit, the game will be shut down.



**Figure 5.3.1:** Win screen



**Figure 5.3.2:** Game over screen

1. References
2. Wiki: <https://en.wikipedia.org/wiki/2048_(video_game)>
3. Github code: <https://github.com/yangshun/2048-python>
4. Link group Github: <https://github.com/serema12/2048-Game>
5. Pygame Learning: <https://www.youtube.com/watch?v=K5F-aGDIYaM&list=PL6gx4Cwl9DGAjkwJocj7vlc_mFU-4wXJq>