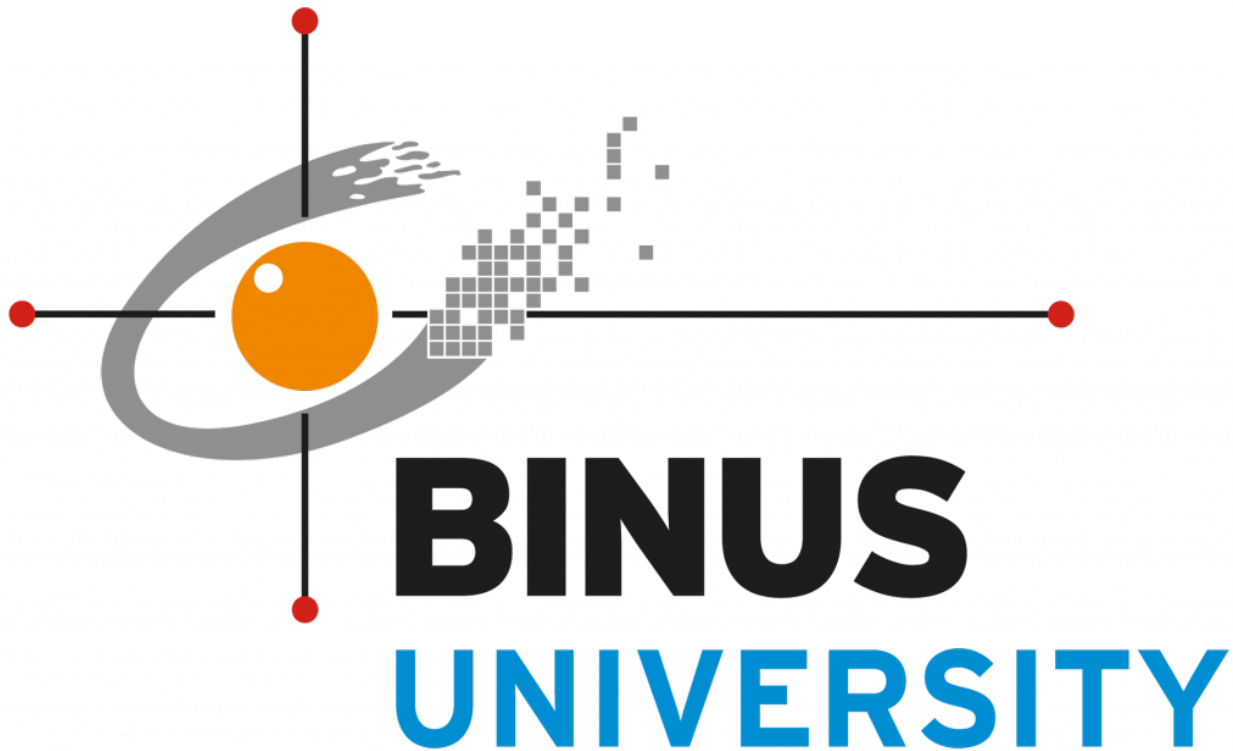


Project Report



By: Alya Nursalma Ridwan

Student Number: 2602184265

Class: L1CC

Course Name: Algorithm and Programming

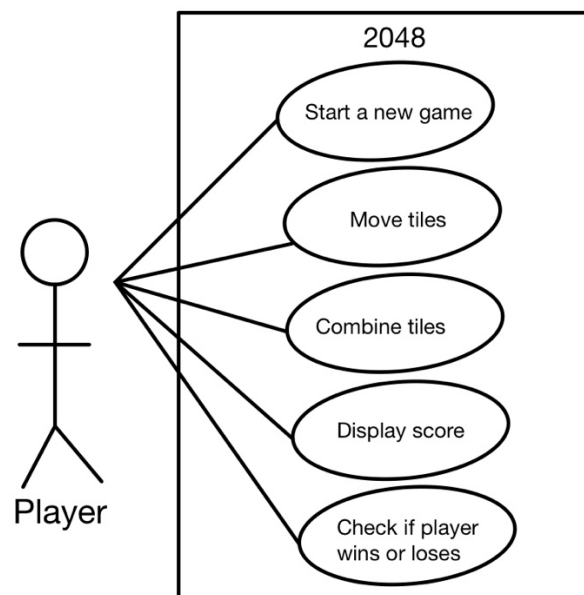
Course Code: COMP6047001

Lecturer: Jude Joseph Lamug Martinez, MCS

Brief Description

The game that I have made is called 2048 which is a single-player game, initially released in 2014 by an Italian developer Gabriele Cirulli. The game is created using a Game class that inherits from the tkinter Frame widget. The class has several methods, including init, which sets up the game and calls other methods to create the game's GUI, bind movement functions, and start the main loop. The class also contains several class variables that define the color scheme and font used in the game. The GUI is created using nested loops to create a 4x4 grid of cells, each of which is a self-contained frame widget. The game allows the user to move the cells using the arrow keys and attempts to combine cells with the same value.

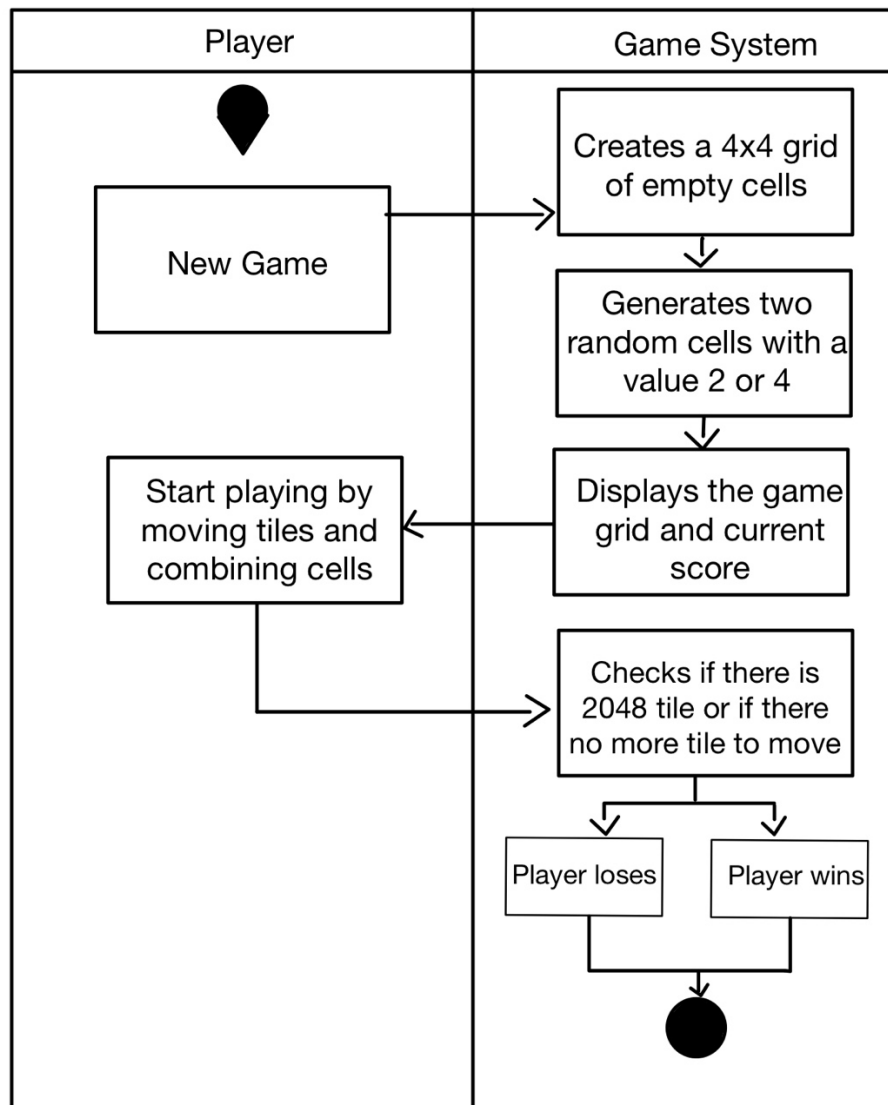
Use-case Diagram



- The player is represented by the actor icon.
- The system is represented by the rectangle with the name "2048 Game"
- The main use cases are:
 - Start a new game: the player initiates a new game by pressing the "New Game" button or choosing "New Game" from the menu.
 - Move the tiles: the player moves the tiles in the grid using the arrow keys

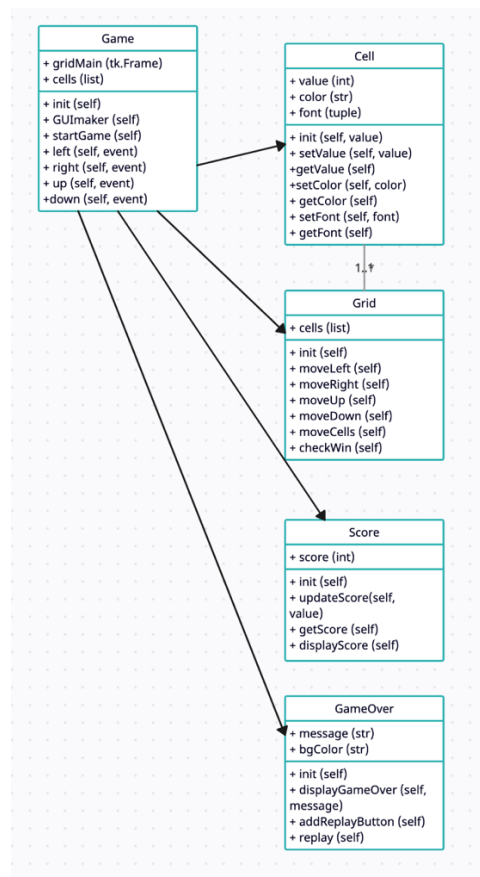
- Combine the tiles: the system automatically combine the tiles with the same number that are next to each other.
- Display the score: the system displays the player's current score.
- Check if the player wins or loses: the system checks if the player has reached the 2048 tile or if there are no more moves available.
- The arrows show the flow of interactions between the player and the system.

Activity Diagram



- The diagram is started by the player choosing "New Game" from the menu or pressing the "New Game" button.
- The system initializes the game grid by creating a 4x4 grid of empty cells.
- The system generates two random cells with a value of 2 or 4.
- The system displays the game grid and the current score on the screen.
- The player can now start playing the game by moving the tiles and combining the cells.
- The flow ends when the player reaches the 2048 tile or when there are no more moves available.

Class Diagram



Modules

In this game I have used two modules which are:

1. **tkinter:** Tkinter is the standard Python interface to the Tk GUI toolkit. It is used to create graphical user interface (GUI) applications. In your game, you have used tkinter to create the game window, grid and other elements of the GUI. This module is used to create the main frame of the game, which is a tk.Frame object, and it's also used to create the grid cells and to set the size, color and other properties of the frames.
2. **random:** The random module is used to generate random numbers and sequences. In your game, you have used random to generate random numbers to place on the grid for the start of the game or when the player makes a move. This module is used to generate random numbers when starting the game, it's also used when the player makes a move to place new numbers on the grid cells.

Essential Algorithms

In this game, there are several essential algorithms that are used to control the game logic:

- **Grid movement algorithm:** This algorithm handles the player's input and moves the tiles on the grid accordingly. When the player makes a move (up, down, left, or right), this algorithm is responsible for shifting the tiles in the desired direction and merging any tiles with the same value.
- **Tile spawning algorithm:** This algorithm is responsible for spawning new tiles on the grid after each move. The algorithm randomly selects an empty cell on the grid and assigns it a value of 2 or 4.
- **Game over detection algorithm:** This algorithm checks if the game is over by checking if there are no more moves available or if the player has reached the 2048 tile. If the game is over, the algorithm displays the game over message on the screen.

- Score calculation algorithm: This algorithm keeps track of the player's score by adding up the values of the merged tiles. The score is displayed on the screen and updated after each move.
- Win detection algorithm : this algorithm check if the player has reached the 2048 tile and display the win message on the screen.

Screenshots of the Game



Lesson Learned/Reflection

In this project, I learned how to create a simple game using Python and the Tkinter library. I learned how to use classes to create a graphical user interface, and how to use the random module to generate random numbers for the game. I also gained a deeper understanding of the game logic and the algorithms used to control the game. I learned how to handle player input, move the tiles on the grid, spawn new tiles, check for game over and win conditions, and keep track of the score. Additionally, I learned about the importance of code organization and commenting, and how it can help to make the code more readable and understandable. Overall, this project provided an excellent opportunity to apply my knowledge of Python and software development to create a functional game, and to learn new skills and techniques along the way.