

INTERIM REPORT 2048 Game

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

– DATA SCIENCE WITH MACHINE LEARNING

By

NAWEED KE

Registration No: 12322524

Section: K23UP

Roll No: 37

15/11/2024 Supervisor

Aman Kumar



School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

October 2024

ACKNOWLEDGEMENT

I NAWEED KE at this moment declare that the research work reported in the dissertation/dissertation proposal entitled “STUDENT MANAGEMENT SYSTEM” in partial fulfilment of the requirement for the award of Degree for Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under the supervision of my research supervisor Mr. Aman Kumar. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith directly complies with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and the highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents an authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

NAWEED KE

REG.NO - 12322524

Table of Contents

- 1 Introduction
 - 1.1 Project Overview
 - 1.2 Purpose and Significance
- 2 Objectives and Scope of the Project
 - 2.1 Project Objectives
 - 2.2 Project Scope
- 3 Application Tools
 - 3.1 Software Applications
 - 3.2 Programming Languages
- 4 Project Structure
 - 4.1 Main Components of the Project
 - 4.2 Classes and Their Functions
 - 4.3 Interaction Between Components
- 5 Flowchart
 - 5.1 Explanation of the Flowchart
- 6 Conclusion

I. INTRODUCTION

1.1 Project Overview

The 2048 Game is a popular single-player puzzle game built on a 4x4 grid. The goal is to merge numbered tiles to create a tile with the value of 2048 or achieve the highest possible score. The game involves strategic decision-making to prevent gridlocks and maximize tile combinations.

Gameplay Mechanics:

- Tiles slide in the chosen direction: up, down, left, or right.
- When two tiles with the same number collide, they merge into a single tile with double the value.
- A new tile (value 2 or 4) appears randomly on the grid after each move.
- The game ends when no valid moves remain.

1.2 Purpose and Significance

This project aims to replicate the 2048 game using Python to:

1. Provide a hands-on opportunity to apply Python programming concepts in a real-world context.
2. Develop algorithmic problem-solving skills.

3. Create a foundation for future exploration in game development.
4. Demonstrate modular programming for improved scalability and maintenance.

II. Objectives and Scope of the Project

2.1 Project Objectives

- Develop a fully functional 2048 game using Python.
- Implement smooth tile movement, merging logic, and score tracking.
- Provide a user-friendly interface for interaction.
- Introduce robust game-over and win conditions.
- Ensure that the game is easy to understand and expandable for future enhancements.

2.2 Project Scope

The project encompasses:

- Designing a text-based or graphical version of the game.
- Building a game logic system to manage tile movements and state updates.
- Encouraging clean coding practices using **Object-Oriented Programming (OOP)** principles.
- Allowing future scalability, such as implementing AI players or larger grid sizes.

III. Application Tools

3.1 Software Applications

- Python 3.x: The primary programming environment.

- IDE Tools:
 - PyCharm for structured development and debugging.
 - VS Code for lightweight coding and quick edits.
- Version Control:
 - Git/GitHub to maintain version history and collaborate.

3.2 Programming Languages of the Project

- python: Chosen for its simplicity, readability, and rich library support.
- Libraries utilized:
 - Tkinter or pygame for GUI development.
 - Random for generating new tiles at random positions.

IV . Project Structure

- Grid System:
 - Handles the initialization of the 4x4 grid.
 - Manages tile placement and updates.
- Game Logic:
 - Processes tile movements, merges, and score updates.

- User Input:
 - Captures directional commands (up, down, left, right).
- Game State:
 - Evaluates the game for win or game-over conditions

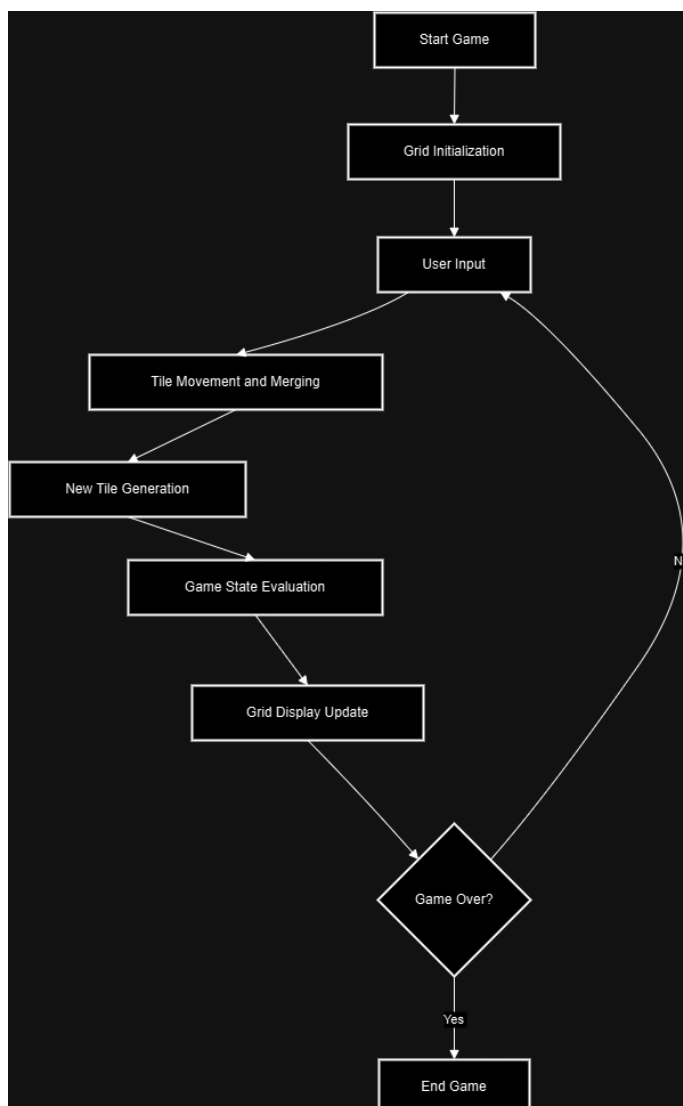
4.2 Classes and Their Functions

- Grid Class:
 - Manages the game board and generates new tiles.
 - GameLogic Class:
 - Handles tile movement, merging rules, and scoring.
 - InputHandler Class:
 - Maps user inputs to corresponding actions.
 - GameState Class:
 - Tracks the state of the game (ongoing, win, or game over).
-

4.3. Interaction Between Components

- The InputHandler receives user inputs and communicates them to GameLogic.
- GameLogic processes these inputs, updates the Grid, and verifies the game state via GameState.
- Grid reflects these updates, displaying the game board and score.

V. Flowchart or Algorithm of the Project



5.1 Explanation of the Flowchart

The game flow is as follows:

1. **Grid Initialization:**
 - The game begins with a 4x4 grid containing two randomly generated tiles (value 2 or 4).
2. **User Input:**
 - Players input a direction (up, down, left, right).
3. **Tile Movement and Merging:**
 - Tiles shift in the input direction, merging adjacent tiles of the same value.
4. **New Tile Generation:**
 - A new tile is added to a random empty spot after each move.
5. **Game State Evaluation:**
 - Checks for a win (2048 tile) or a loss (no valid moves).
6. **Grid Display Update:**
 - Updates the grid and score display after every move.
7. **Repeat:**
 - Steps 2–6 repeat until the game ends.

6. Conclusion

This project demonstrates how Python can be used to implement a classic game like 2048. By integrating modular design, clear logic, and user interaction, the project highlights the application of key programming concepts and lays a foundation for exploring more advanced topics such as graphical user interfaces and AI-driven automation.

The 2048 game project not only achieves its objectives but also provides an enjoyable and educational experience for both developers and players.