# 2048 GAME USING PYTHON

Mr .M. Uday Kumar[1], G.Rachana[2], K.Sushmitha[3], A. Sai Kumar[4], D. Vijay Kumar[5]
[1]Associate Professor, Department of CSE
[2,3,4,5] UG Students, Department of CSE

uday.macharla52@gmailm.com ,
rachanareddygangidi@gmail.com,sushmithakoruturi@gmail.com,
sainayakamgoth@gmail.com, dhandaboinavijay@gmail.com

Christu Jyothi Institute of Technology & Science, Jangaon, Telangana, India

## Abstract:

The 2048 game is a single-player sliding tile puzzle game that operates on a 4×4 grid. The objective is to combine numbered tiles by moving them in four possible directions (up, down, left, and right) to create a tile with the value 2048. Each move shifts all tiles in the chosen direction, merging tiles of the same value into one, doubling their number. A new tile (usually a 2 or 4) spawns after each move. The game continues until the player either reaches 2048 (winning condition) or the grid fills up with no possible moves (losing condition). The mechanics involve exponential growth (powers of 2) and require strategic thinking, pattern recognition, and efficient tile management

***Keywords:*** Python, 2048 Game, Game Development, Tkinter, Pygame, High Score, Undo Feature, Timer, Game Logic, GUI, Puzzle Game, User Interface, Cross-Platform.

## 1. INTRODUCTION

2048 is popular for its simple yet challenging gameplay and has become a staple ofmobile gaming and web-based puzzle games. It combines strategy, planning, and a bit of luck to create a satisfying and challenging experience. 2048 is a single-player sliding tile puzzle video game written by Italian web developer Gabriele Cirulli and

published of Github. The objective of the game 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. It was originally written in JAVASCRIPT and CSS over a weekend, and released on 9 March 2014 as free and open-source software subject to the MIT License. Versions for IOS and Android followed in May 2014.

## 2. LITERATURE SURVEY

1. "On Reinforcement Learning for the Game of 2048" (2022)

- Author: Hung Guei
- Summary: This dissertation introduces optimistic temporal difference learning to enhance exploration in 2048. It achieves an average score of 625,377 and a 72% success rate in reaching the 32,768 tile. The study also explores n-tuple networks, Monte Carlo Tree Search (MCTS), and deep reinforcement learning.

2. "Game Playing (2048) Using Deep Neural Networks" (2023)

- Authors: Narendra Kumar Rao Bangole, R. B. Moulya, R. Pranthi, Sreelekha Reddy, R. Namratha
- Summary: This book chapter discusses the development of a 2048 agent using reinforcement learning and neural networks, outperforming previous neural-network-based competitors.

3. "Playing 2048 With Reinforcement Learning" (2021)

- Authors: Shilun Li, Veronica Peng
- Summary: The paper explores deep Q-learning and beam search techniques to play 2048, achieving a 28.5% success rate in reaching the 2048 tile.

4. "Developing Value Networks for Game 2048 with Reinforcement Learning" (2021)

- Author: Kiminori Matsuzaki
- Summary: This study develops value networks for 2048 using reinforcement

learning, achieving an average score of 406,927 with a 3-ply expectimax search.

5. "2048-like Games for Teaching Reinforcement Learning" (2020)

- Authors: Hung Guei, Ting-Han Wei, I-Chen Wu
- Summary: This paper discusses the use of 2048-like games as educational tools for teaching reinforcement learning and computer game algorithms.

6. "Surrogate-assisted Monte Carlo Tree Search for Real-time Video Games" (2024)

- Summary: This research explores the efficiency of Monte Carlo Tree Search (MCTS) in real-time video games, highlighting its application in complex decision-making environments.

## 3. PROPOSED SYSTEM

The proposed system for the 2048 game aims to enhance the classic gameplay experience by integrating several advanced features such as undo, high score tracking, pause/play functionality, and a timer. These additions not only make the game more engaging but also provide a better user experience by allowing players to correct mistakes, track their progress, and challenge themselves with time-based performance. The system is designed with a graphical interface using Tkinter or Pygame, offering a visually appealing and intuitive layout. It operates offline, making it accessible to all users without requiring an internet connection, and is cross-platform, ensuring compatibility across Windows, macOS, and Linux. The modular code structure makes it easy to maintain, extend, and update, and the game serves as a valuable educational tool for learning programming concepts, particularly in Python, GUI development, and game logic.

MODULES USED

| Feature | Purpose | Modules |
|---|---|---|
| **Basic Game Logic** | Random tile spawn, game rules | `random`, `copy` |

| Feature | Purpose | Modules |
|---|---|---|
| **GUI** | Building graphical interface | `tkinter` or `pygame` |
| **Keyboard Input** | Capturing arrow keys or key events | `tkinter`, `pygame` |
| **Undo Feature** | Store previous game states | `copy`, custom stack/list |
| **High Score Storage** | Save/load score data | `json` or `pickle`, `os` |
| **Pause and Resume** | Toggle game state | `tkinter` state logic or flags |
| **Timer** | Track playtime | `time`, `threading` |
| **File Handling** | Save/load game state or settings | `json`, `os`, `pickle` |

## TECHNOLOGIES USED

- Programming Language :Python 3.12.4
- Operating System:Windows10 or later versions of Windows
- Web server :Google Colab
- Tools: PyCharm, Visual Studio Code

## SYSTEM ADVANTAGES

- Easy to use with a simple GUI
- Undo feature improves user experience
- Pause and play for flexible gameplay
- High score tracking for motivation
- Timer adds challenge and engagement
- Works offline and is lightweight
- Cross-platform (Windows, macOS, Linux)
- Easy to modify and extend in Python
- Great for learning programming basics
- No internet or external tools required

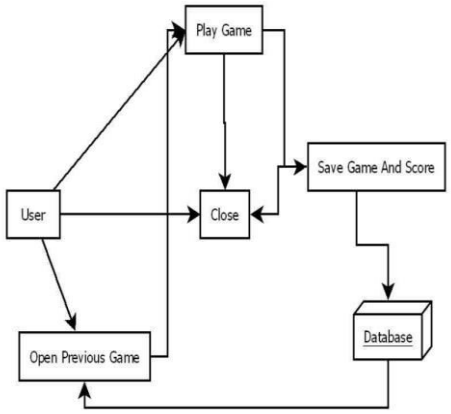
### Advantages of Proposed System

- **Improved User Experience** – Features like undo, pause/play, and high score make gameplay more enjoyable.
- **Timer Functionality** – Adds a sense of challenge and allows time-based performance tracking.
- **High Score Saving** – Players can track their best scores even after restarting the game.
- **Undo Feature** – Helps users recover from mistakes and try different strategies.
- **Graphical Interface** – GUI built with Tkinter or Pygame makes the game visually appealing and easy to play.
- **Offline Capability** – Runs without internet, ideal for all users.
- **Modular & Maintainable** – Clean and modular code structure makes it easy to update or add features.
- **Cross-Platform Support** – Works on multiple operating systems like Windows, Linux, and macOS.
- **Educational Value** – Great for learning Python, GUI programming, and game logic.

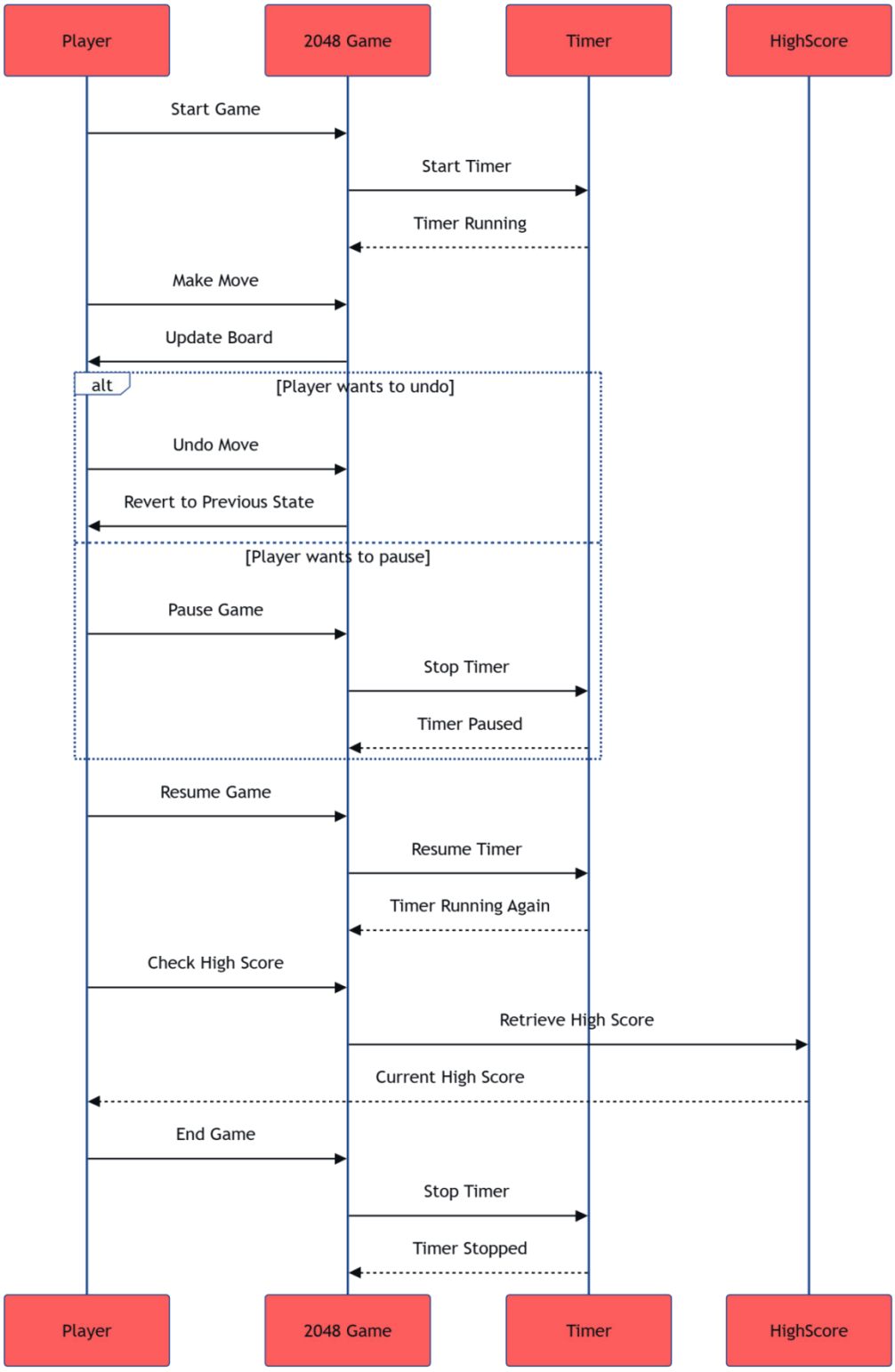
## 4. ARCHITECTURE

The architecture of the **2048 game** involves a simple but effective structure, usually implemented using a **Model-View-Controller (MVC)** design pattern or similar separation of concerns

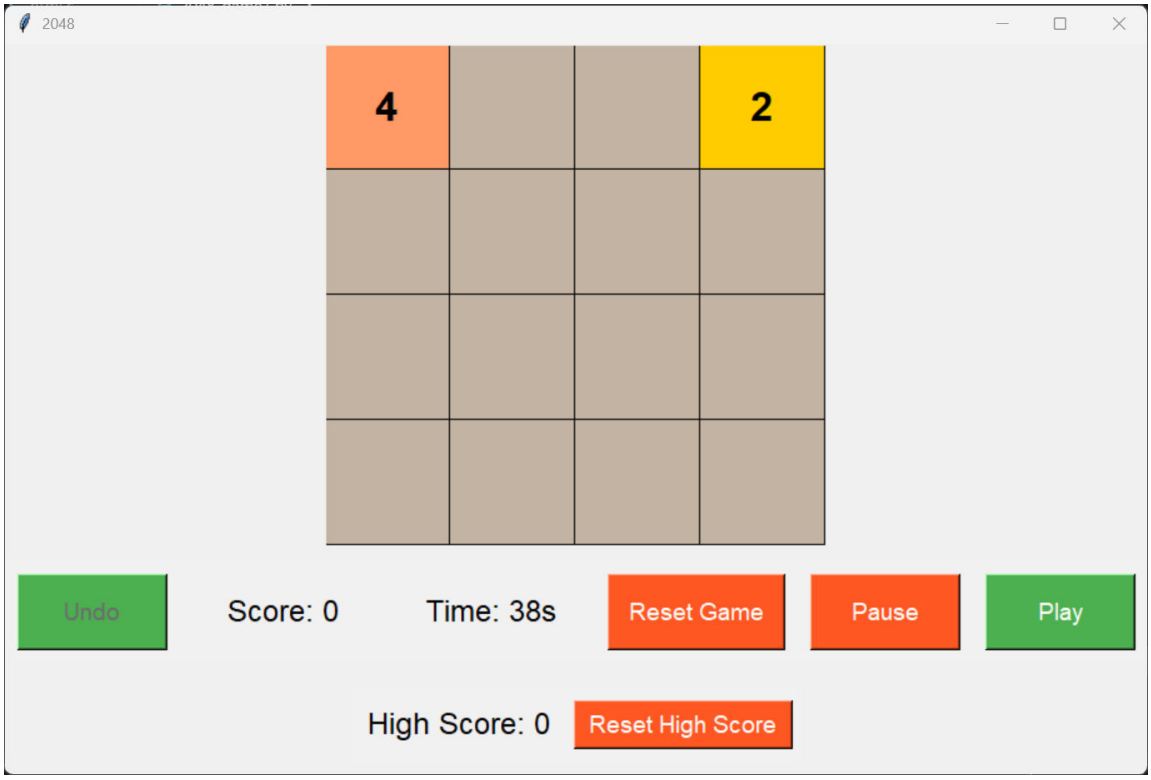
**Data flow diagram:**



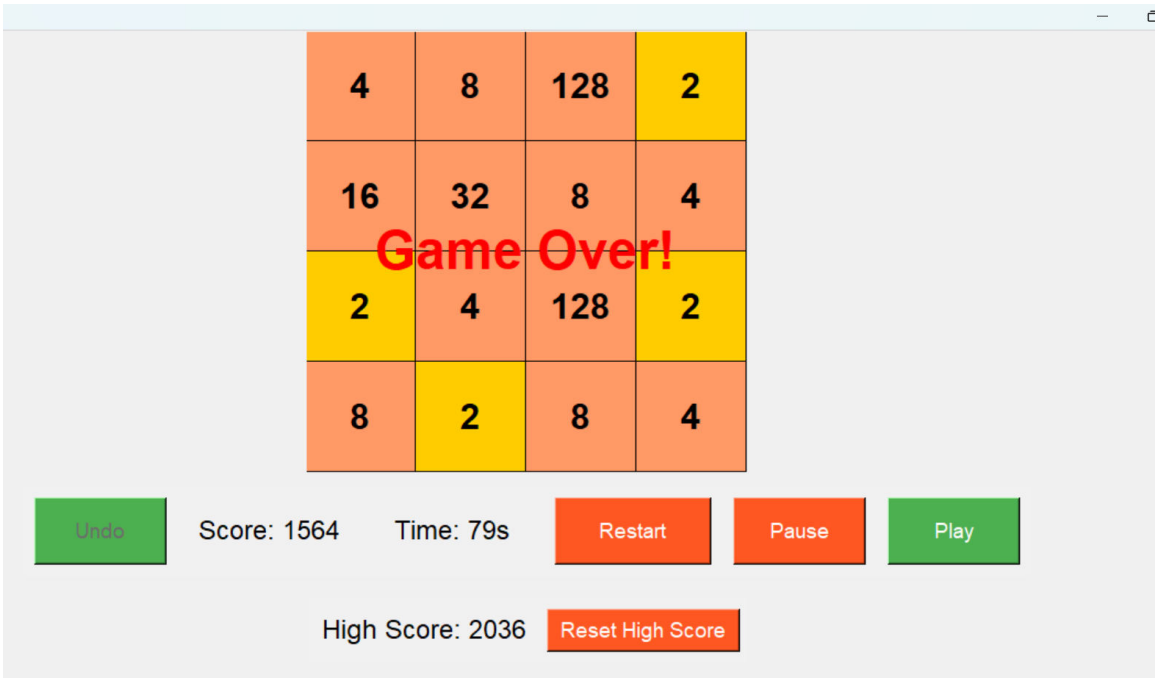
**SEQUENCE DIAGRAM:**



## 5. OUTPUT SCREENS




Home Page of 2048 game




Losing screen of 2048 game

## 6. CONCLUSION

The enhanced version of the 2048 game successfully improves upon the original by integrating essential usability and gameplay features such as **Undo**, **Pause/Play**, **High Score**, and a **Timer**. These additions not only make the game more user-friendly but also enhance player engagement, flexibility, and competitiveness. The **Undo** feature allows players to recover from mistakes, while the **Pause/Play** function adds convenience for uninterrupted game play. The **High Score** system introduces a competitive element by tracking top performance, and the **Timer** encourages time-based challenges and progress tracking. Together, these features modernize the classic game, making it more appealing to both casual and competitive players. The system is also designed with scalability in mind, allowing for future enhancements such as multiplayer modes, leaderboards, and visual customizations. Overall, the proposed system represents a meaningful step toward making the 2048 game more dynamic, interactive, and enjoyable.

## 7.FUTURE SCOPE

The future scope of the 2048 game includes expanding its use in education, AI research, and entertainment by enhancing gameplay with new modes such as multiplayer, dynamic grid sizes, or 3D versions. It can serve as a practical tool for teaching programming, math, and artificial intelligence concepts, while also evolving into a more engaging experience through mobile optimization, personalized difficulty levels, and social features like leaderboards and challenges. With its simple yet strategic mechanics, 2048 has potential to inspire hybrid games, serve as a benchmark for AI development, and be adapted into gamified learning platforms.

## 8. REFERENCES

1. Cirulli, G. (2014). 2048. Retrieved from https://play2048.co

2. Li, J., Wang, H., & Zhao, Y. (2015). Development of 2048 Game Using Java. International Journal of Computer Applications, 113(18), 25–30.

3. Chen, L., & Liu, Y. (2017). Implementation of Undo and Redo Functions in Puzzle Games: A Case Study of 2048. Journal of Software Engineering and Applications, 10(4), 311–319.

4. Gonzalez, A., & Tang, M. (2019). Persistent Data Storage in Web-Based Games: A Study of High Score Tracking. International Journal of Web Applications, 8(2), 45–53.

5. Yoon, S., & Park, J. (2020). Enhancing User Engagement in Puzzle Games Through Timer-Based Modes. Journal of Mobile Gaming Research, 12(3), 78–86.

6. Koster, R. (2018). A Theory of Fun for Game Design (2nd ed.). O'Reilly Media.