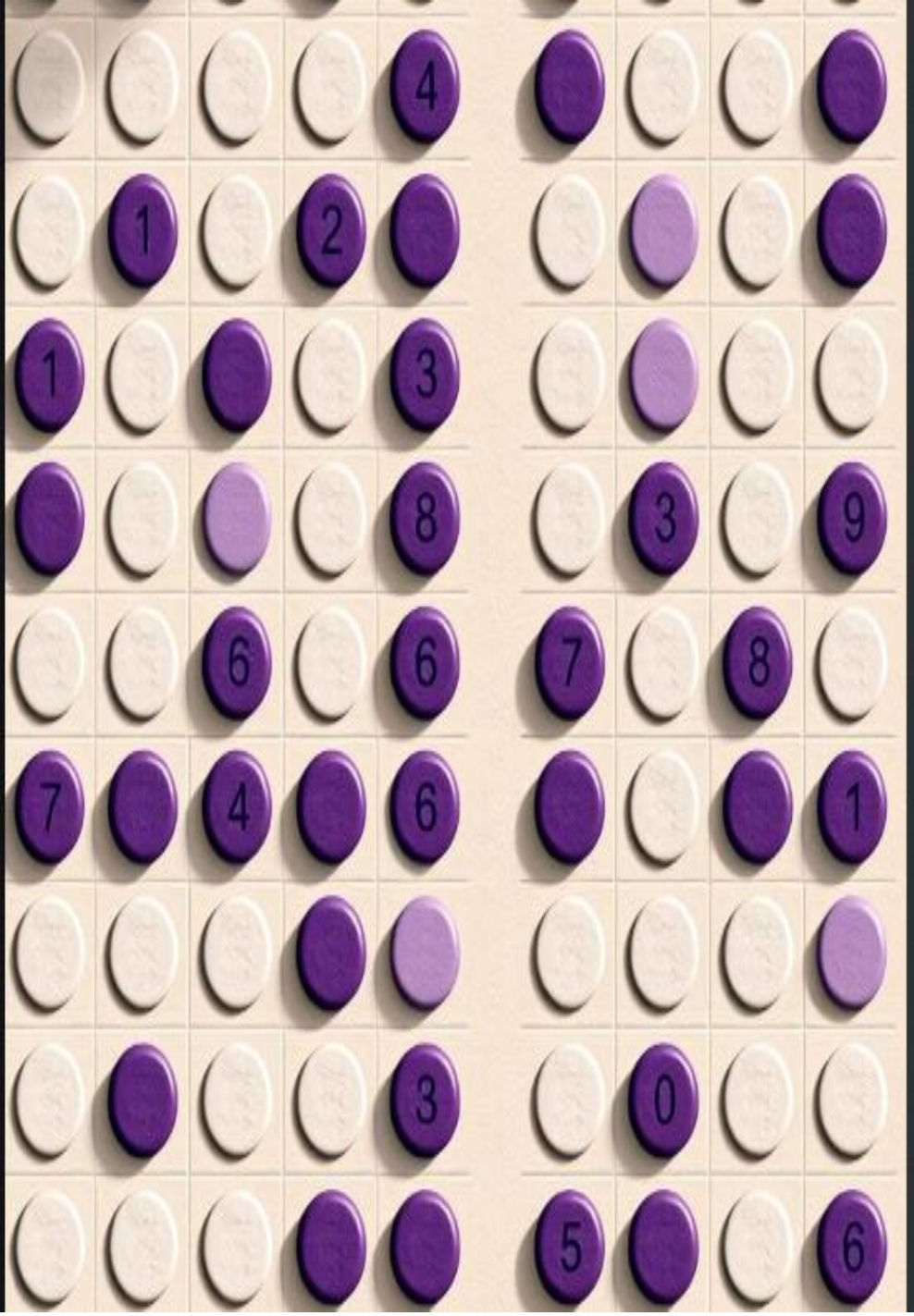


Minesweeper: A Python Project

Minesweeper is a classic puzzle game where you click on squares in a grid, trying to avoid hidden mines. It's all about using logic to figure out where the mines are. In this project, we'll create a Minesweeper game using Python

The Minesweeper Challenge

Objective

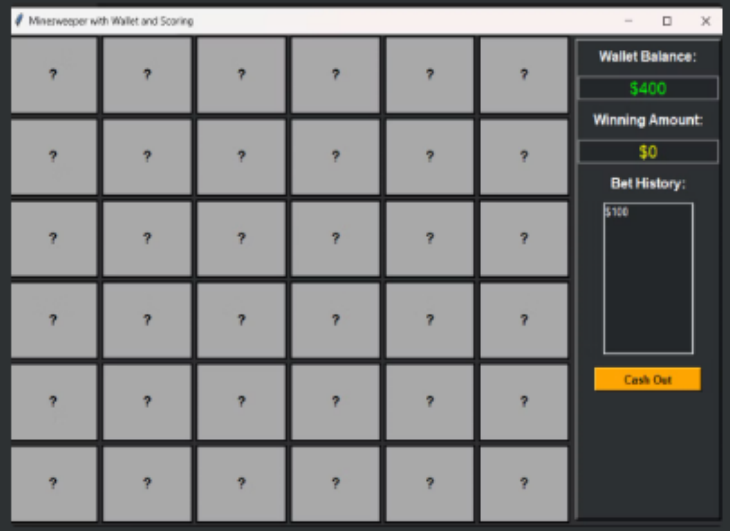
The goal is to hit a all non-mine cells on the grid without detonating any mines.

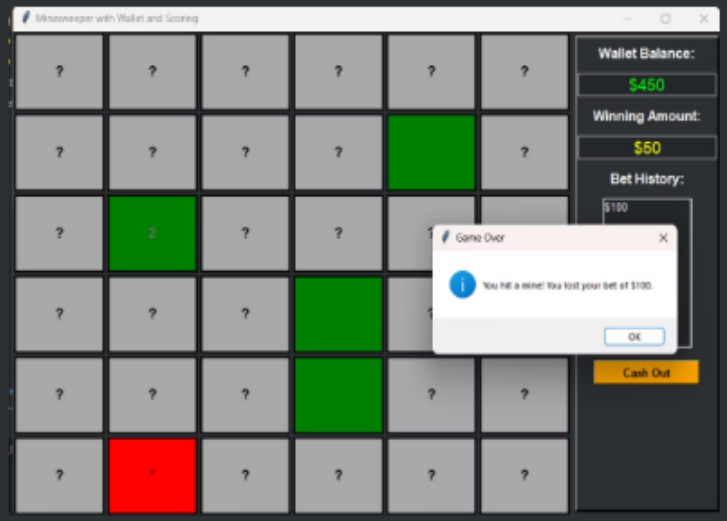
Gameplay

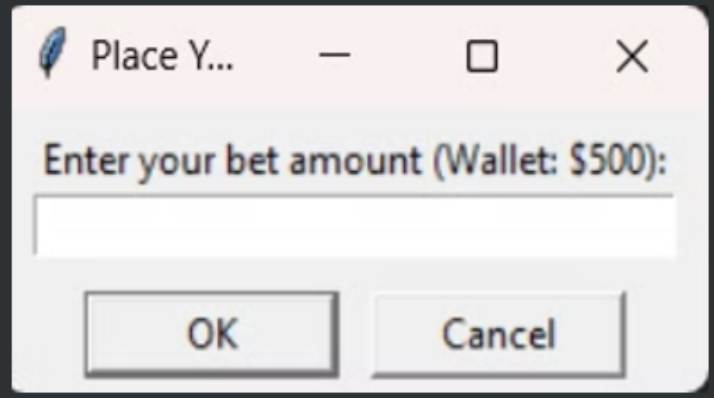
Players must use logic, pattern recognition, and calculated risk-taking to identify safe cells and mark suspected mines.

Tension

With every click, the excitement grows as you carefully uncover the grid while trying to avoid hitting a hidden mine. One wrong move, and it's game over.







Building the Minesweeper Grid

Grid Size

The game board will be a 6x6 grid.

Mine Placement

Mines will be randomly scattered throughout the grid.

User Interaction

Players will be able to place bets and start the game, testing their minesweeping skills.

Key Libraries and Tools

Python

this programming laungage super flexible and has tons of helpful libraries to make coding easier.

Tkinter

The standard GUI library for Python, enabling the creation of the game's visual interface.

Random

Python's random module to scatter the mines randomly across the grid, so every game feels fresh and unpredictable.

Scoring and Victory

1 Scoring

Scoring Points

- **Starting Condition**: Players must place a bet of at least $10 to access the game.
- **Correct Guess**: If a player successfully avoids the mines and wins the game, their bet amount will be added to their wallet balance (or multiplied, depending on your rules).
- **Incorrect Guess**: If the player loses, the bet amount is deducted from their wallet balance.

2 Victory Conditions

- **Clearing the Grid**: The player wins if they reveal all safe squares without hitting a mine.
- **Cashout Option**: Players can choose to cash out their winnings after a successful game or at any point allowed by the game.