

# S.P.A.R.K.: Strategic Probabilistic Analysis of Real-time Minesweeper Kinetics

## 1 Introduction

This Python script implements a Minesweeper game with additional functionality to display adjacent probabilities. It utilizes the `tkinter` library for the graphical user interface.

## 2 Game Overview

SPARK enhances the traditional Minesweeper gameplay by providing strategic probabilistic analysis of neighboring cells. When a cell is opened, SPARK calculates the probability of neighboring cells containing mines, thus providing the user with the percentage chance of a mine in the unopened cell. This analysis gives the user an edge by allowing them to make informed decisions based on probability.

## 3 How to Use

### 3.1 Launching the Game

1. Ensure you have Python installed on your system.
2. Run the script using Python. You can launch the game in the following ways:
  - (a) Open a terminal or command prompt, navigate to the directory containing the Python script, and run the script using the command:

```
python filename.py
```

Replace `filename.py` with the name of your Python script.

- (b) Open IDLE (Integrated Development and Learning Environment), open your Python script, and run the script by selecting **Run > Run Module** from the menu.

- (c) Use Python IDEs (Integrated Development Environments) like PyCharm, Visual Studio Code, etc. Open your Python script in the IDE and run the script by clicking the "Run" button or using the corresponding keyboard shortcut.

The game window will appear after running the script.

### 3.2 Gameplay

- Left-click on a cell to reveal its content.
- The game ends if you click on a mine, revealing all mines, or when you have successfully revealed all non-mine cells.

## 4 Game Features

- **Adjacent Probabilities:**
  - The game calculates and displays the probability of adjacent cells containing mines when a cell is revealed.
  - The probabilities are shown as percentages on the buttons.
- **Auto-Clearing Empty Areas:**
  - When an empty cell is revealed, all adjacent empty cells will be automatically cleared, revealing their content.

## 5 Class Structure

- **Cell Class:**
  - Represents each cell on the game board.
  - Tracks its open status, probability array, neighboring blocks, and more.
- **Minesweeper Class:**
  - Manages the game mechanics.
  - Initializes the game board, places mines, reveals cells, updates probabilities, and handles user interactions.

## 6 Functions

- **create game board:**
  - Sets up the graphical user interface by creating buttons for each cell on the game board.

- **initialize game:**
  - Initializes the game by setting up the mines and neighbor relations.
- **reveal cell:**
  - Reveals the content of a cell when clicked by the user.
  - Handles mine explosions and auto-clearing of empty areas.
- **update probabilities:**
  - Calculates and updates the probabilities of adjacent cells containing mines when cells are revealed.

## 7 Additional Notes

- The game automatically ends when all mines are successfully revealed or all non-mine cells are revealed.
- The script is designed to handle left-clicks to reveal cells.
- Probabilities are updated dynamically as the game progresses.

## 8 Dependencies

- Python 3.x
- Tkinter (usually included in Python installations)

## 9 Development Environment

This script was developed and tested in a Python environment.

## 10 Disclaimer

This game is for entertainment purposes only and does not guarantee the accuracy of probabilities or the absence of bugs.

## 11 Authors

Aditya Kumar B22ME004  
 Sonal Raj B22ME062  
 Yash Golani B22ME072  
 Sangameshwar Wanole B22CH029