M’Sweeper

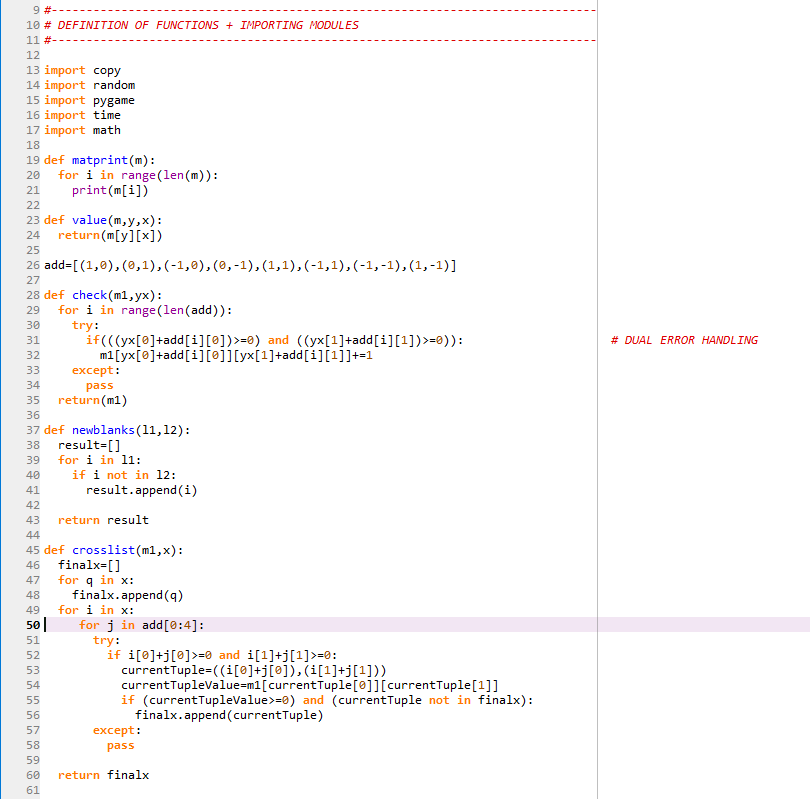
-Aman Thukral

M’Sweeper is an adaptation of Minesweeper, made using python and a module known as pygame. The main aim of the game is to put a flag on all the mines, without revealing them. The numbered cells help with this, as they indicate the number of mines in a 3x3 space around them.

Controls:

**Mouse** to move about the cells

**Left Click** to reveal the cell

**Right Click** to plant a flag to scout mines

Modules imported

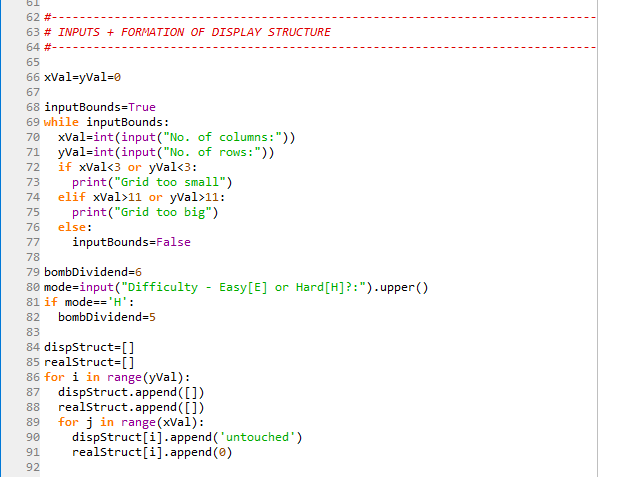
Prints Structure comprehensibly

To give value at a certain coordinate

Handles numbering of cells around a mine

Prints distinct elements between two lists

Gives those coordinates around a certain provided coordinate, where it doesn’t correspond to a mine



**Minimum Grid:** 3x3

**Maximum Grid:** 11x11

Difficulty decides number of bombs

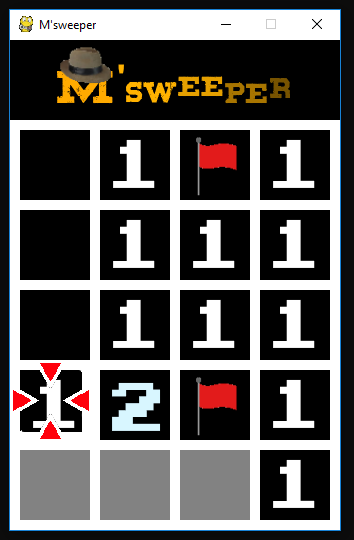


Bombs are proportional to grid size and randomly placed

The blank cells of a region are placed in a list and hence “grouped” so that they may all reveal at the same time during execution

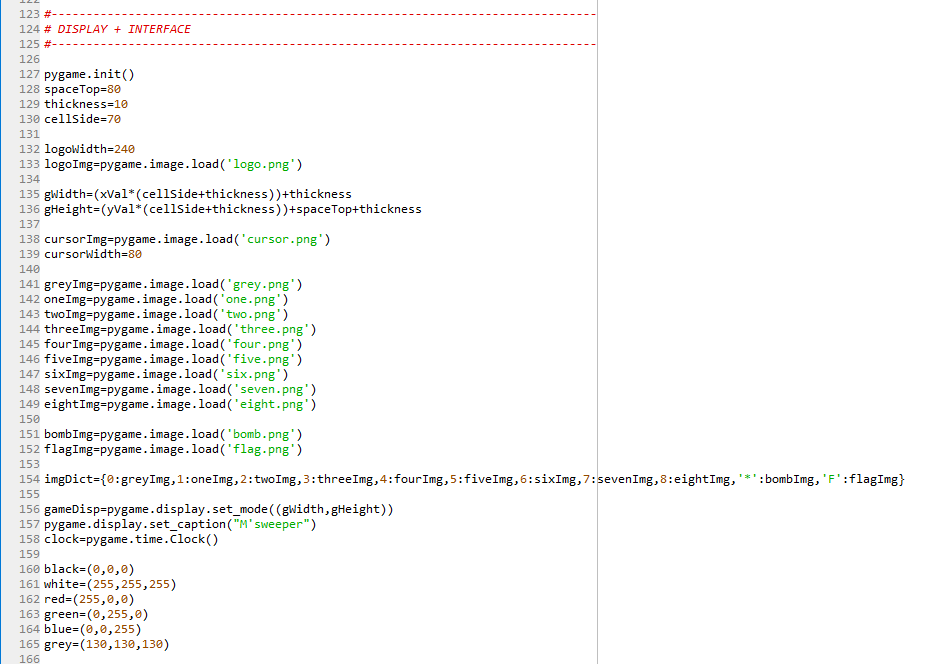
PYGAME

The Pygame Library is used to give a visual experience and aesthetic to the game. For the visual presentation, many assets were created and implemented in the program, including a multitude of images and fonts.



Logo, cursor, flag – **Self Created Assets**

Fonts - **Downloaded**



Setting up game display

Loading of assets and setting values for constants

RGB colour encoding



Determining display coordinates

Printing cursor

For printing text

Determining cell coordinates

Game Loss procedure

Game Win procedure

Event Handling

Event Handling is a major part of the code, as it is the core facet of displayed structure – real structure interaction.



Handling **Left Click:**

Revealing values of cells

Handling **Right Click:**

Flagging cells which are potential mines



Creating the display

**PYTHON CONCEPTS USED:**

60 fps

Win Condition

* Lists

MODULES USED:

* Pygame
* Copy
* Random
* Time
* Math
* Tuples
* Dictionaries
* Loops and If statements
* Defining functions