**VIETNAM NATIONAL UNIVERSITY**

**INTERNATIONAL UNIVERSITY**

A picture containing text, room, gambling house

Description automatically generated

**DATA STRUCTURES AND ALGORITHMS PROJECT REPORT**

***Project: Minesweeper***

***Language: Python***

***Course: DSA – Semester 2 (2021-2022)***

***Link:*** [DuyVu285/DSAProject: The DSA project- Minesweeper Game (github.com)](https://github.com/DuyVu285/DSAProject)

***Group:***

**Vũ Nhật Duy – ITITIU17047**

1. ***Introduction***
2. ***What is Minesweeper***

Minesweeper is a single-player puzzle video game. The objective of the game is to clear a rectangular board containing hidden "mines" or bombs without detonating any of them, with help from clues about the number of neighboring mines in each field.

1. ***Difficulties in creating Minesweeper***

The program is designed from scratch with the help of libraries in python. Thus, learning how to design the game is time-consuming. However, it rewards great satisfaction in coding.   
The undo function is a feature that is required in the project. Because the function must cover every possible move that player can make, the coding process takes quite a considerate amount of time and lots of thinking.

1. ***Scope***

This report represents the overall result of our group with the Minesweeper project. It illustrates how Minesweeper is made such as user interface, algorithms, data structures and features.

1. ***Classes***

***The classes:***

**cell.py**: The design of cells with their functions(cell constructor, show cell, click actions,..)

**gamewindow.py**: The design of game window which is responsible what player can see and interact(grid, labels, losing, winning, clicking,...).

**settings**.**py**: The settings of the game(width, height,...).

**utils**.**py**: Simple width/length function for ease of usage.

**startmenu**.**py**: The design of a start menu window for more user experience.

**main**.**py:** This file acts as the point of execution of the program.

1. ***Usage of Learned Lectures***
2. ***How to play Minesweeper***

- **The principles behind Minesweeper:** each Minesweeper game starts with a grid of unmarked squares(or in this case we call it cells). After clicking one of these squares, it is either a safe cell which holds a number or a “mine” cell that indicates player’s loss. It's player job to make use of the numbers to figure out which of the blank cells have mines and which are safe to click.

- **Mouse's left and right buttons**: the mouse is the only tool that player needs in order to play Minesweeper. The left mouse button is used to click squares to show whether it is a mine or a safe square, while the right mouse button is used to flag cells that contain mines.

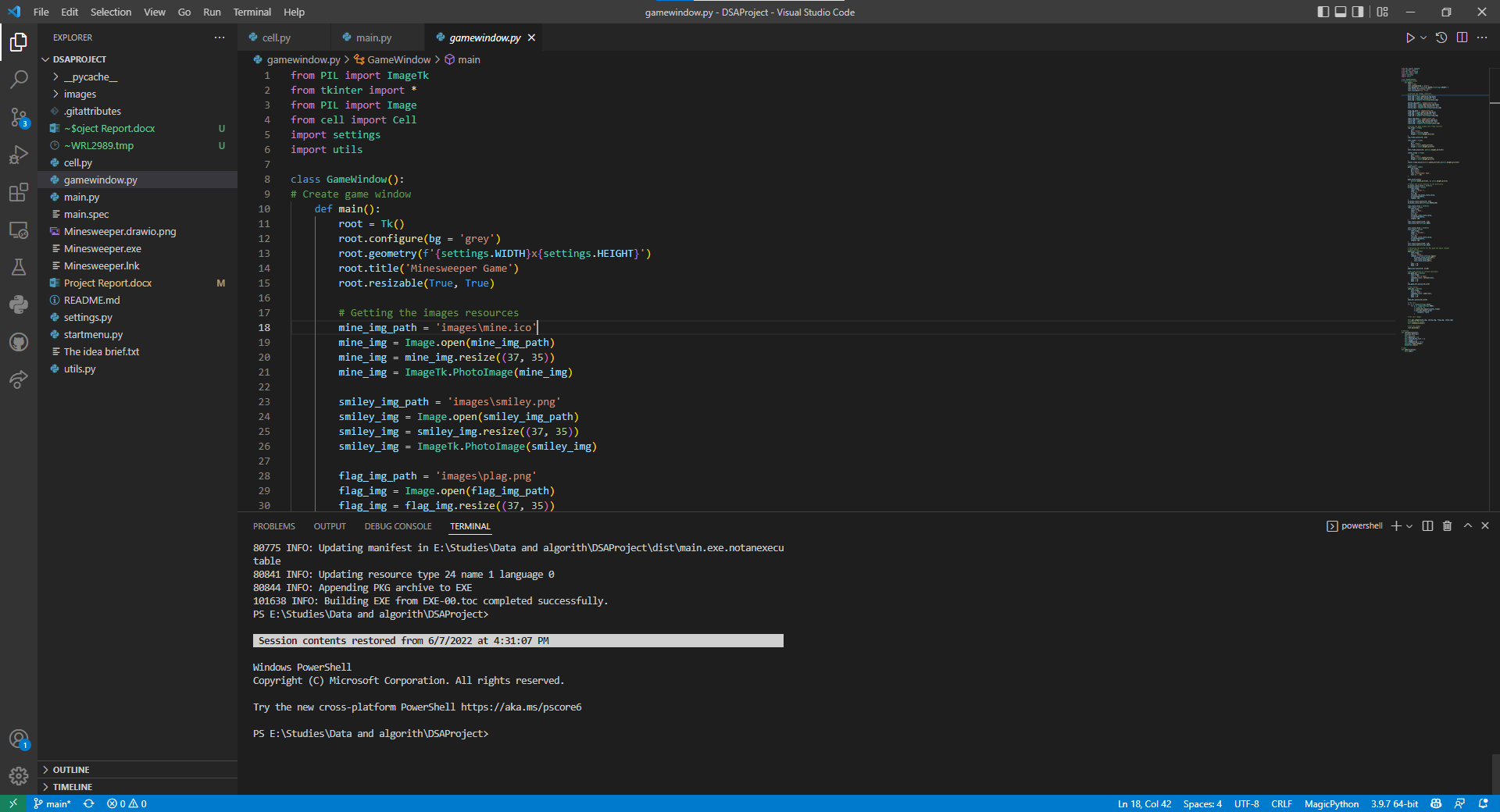
- **The meaning of the numbers**: a number on a cell refers to the number of mines that are currently neighboring that cell. For example, if there the cell has the number 1 on it, you know that the cells surrounding it has a mine hidden beneath one of them.

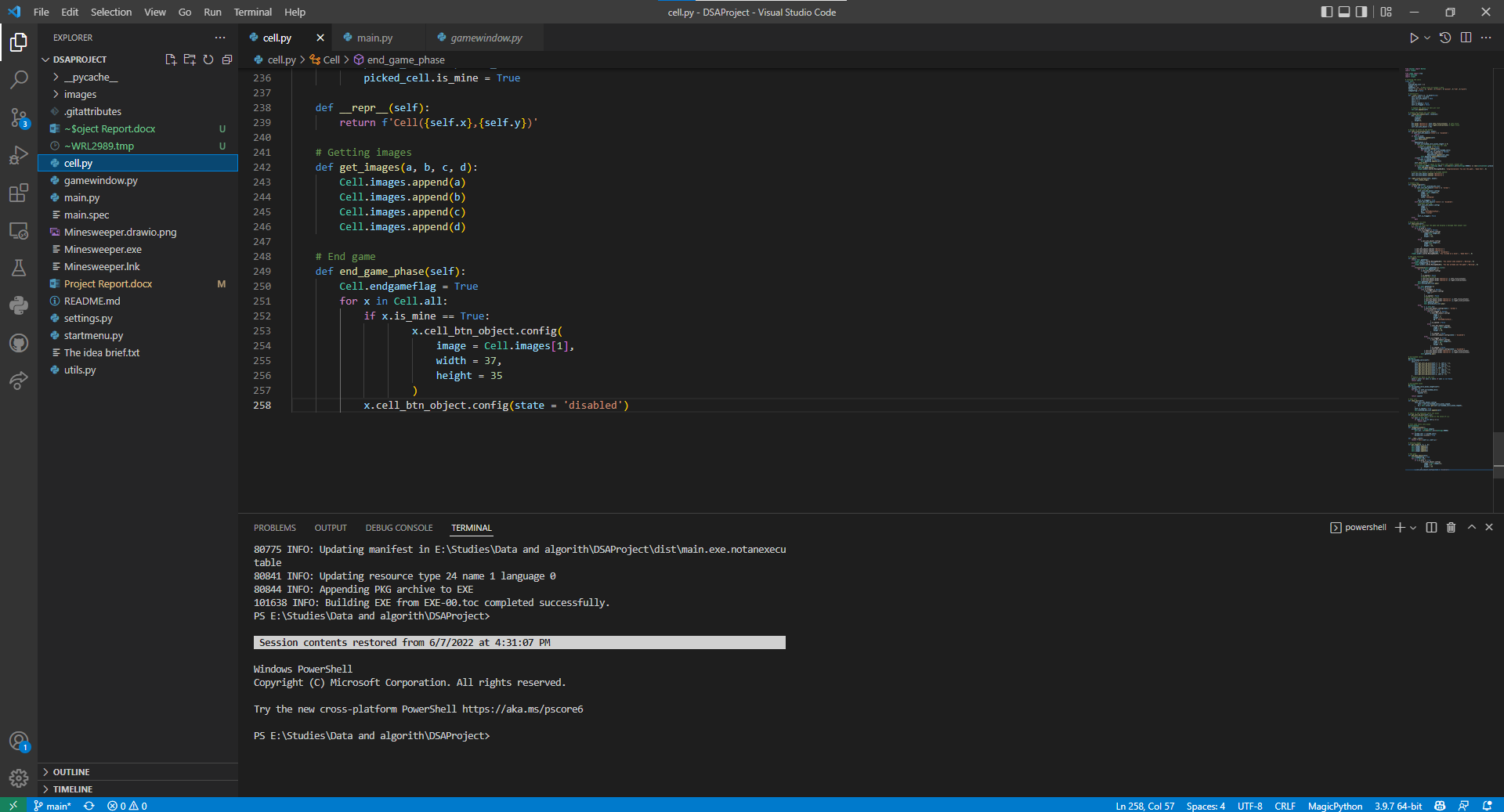
- **Win condition**: player have clicked all the safe cells without triggering any mines.

1. ***Algorithms and Data structures***

***The code***: [DuyVu285/DSAProject: The DSA project- Minesweeper Game (github.com)](https://github.com/DuyVu285/DSAProject)

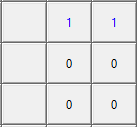
Some images of the code:



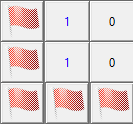


***Features:***

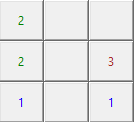
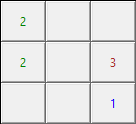
**-** **Clear unnecessary cells:** When the left click event occurs, the open property is set to true, and the surrounding mines are also checked. If the value is 0, then use **recursion** to open the surrounding cells. The recursion stops when the value is not 0. For example, the player hits a 0 cell, unimportant cells are opened. To find the code, find **surrounded\_cells functions in cell.py**.



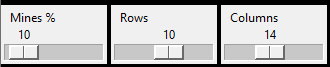
**-** **Flagging mines**: If player is not sure of a specific cell, you can flag with right-click to avoid it until later. Player can undo by right-clicking it once again to make it becomes a normal cell. To find the code, find **create\_flag function in cell.py**.



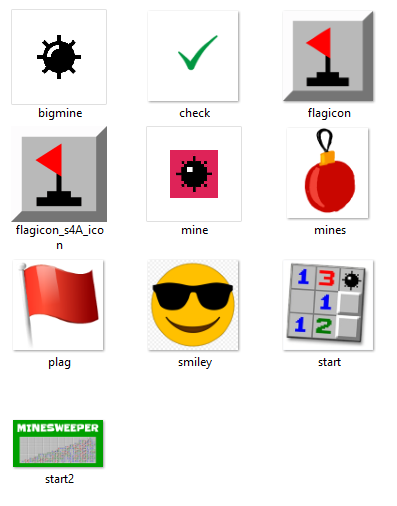
**- Undo feature:** For undo feature, the properties of **Stack** are considered, which take advantages of the “last in first out” property to undo the player’s most recent steps by “popping” it out of the Stack. When the user left-clicks a number or an empty cell, the "push" method is used to add step to the Stack, which reflects the player's most recent move. The player can undo an unlimited number of movements for any sort of move, including clicking on marked cells, empty cells, and neighbor cells. Especially, the player can even loss games. As the game is coded with python, list offers similar functions to a stack which is used in the game. To find the code, find **undo function in cell.py**. The images below are before and after redo.



**- Difficulties:** The game offers a variety of difficulties by choosing the number of rows and columns it can created. Also, the percentage of mines can also be adjusted for more challenges.To find the code, check **buttons functions in window.py**.



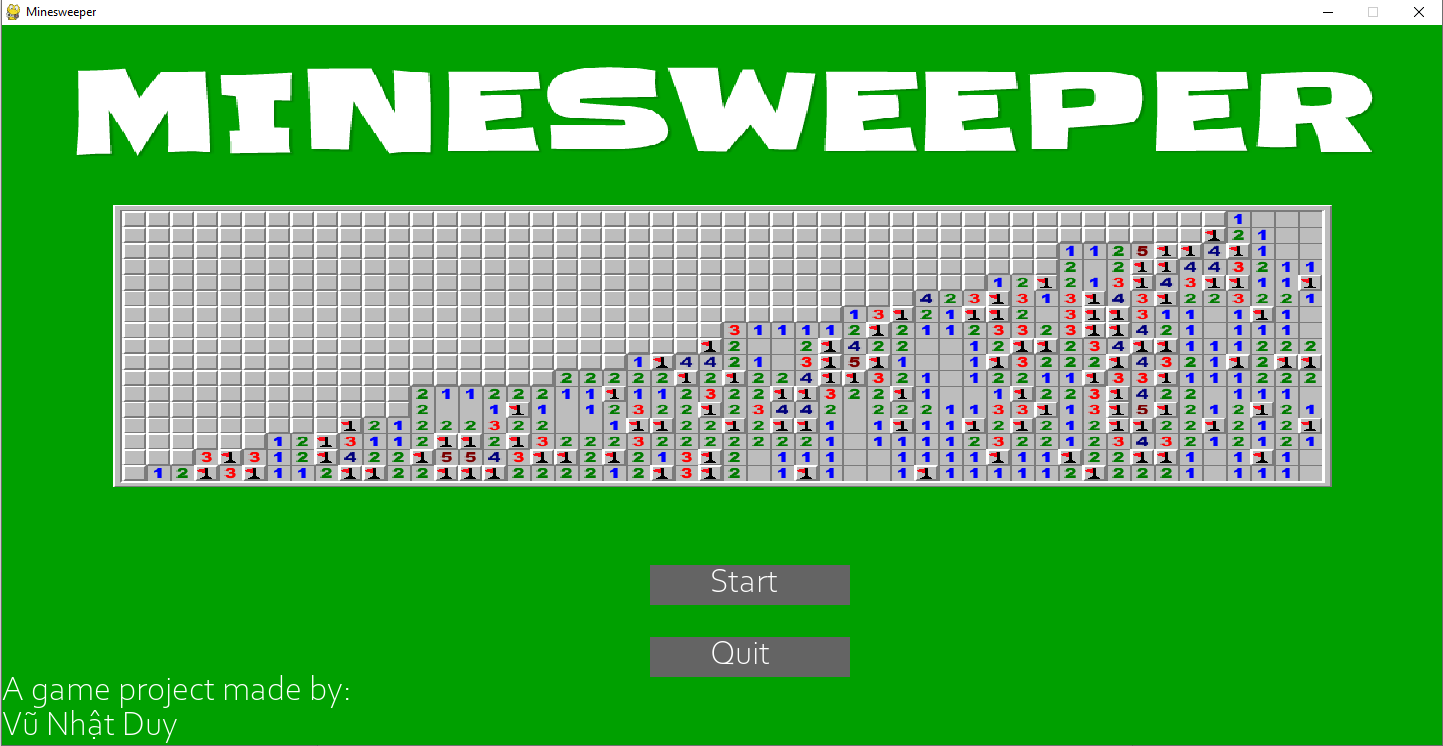
- **Images:** The images below are in images folders. Some of these are implemented in the game, offer a better visual in player’s game experience. Some are unused for various reasons.

******

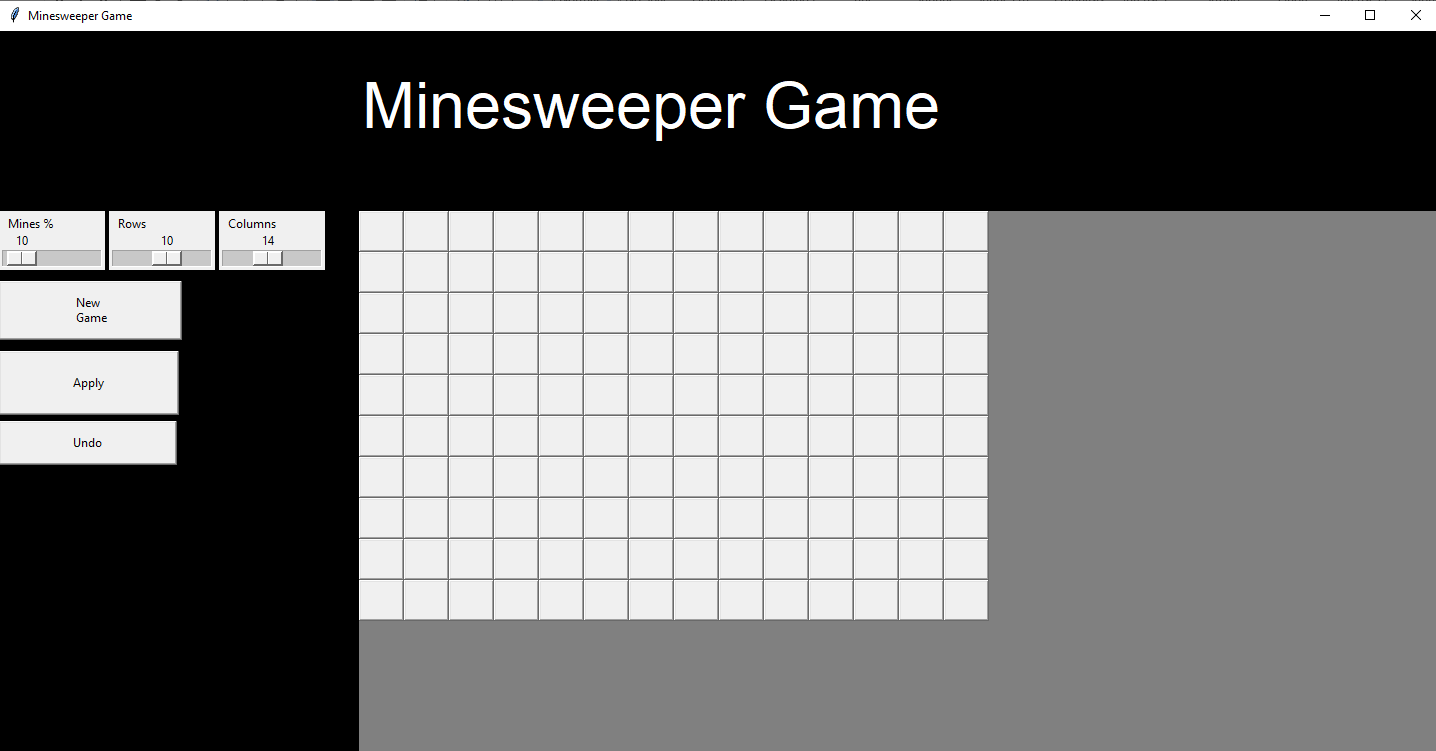
1. ***User interface***

These are what player can see and interact:

**Start menu:**



**Game window:**



1. ***Further implementation***

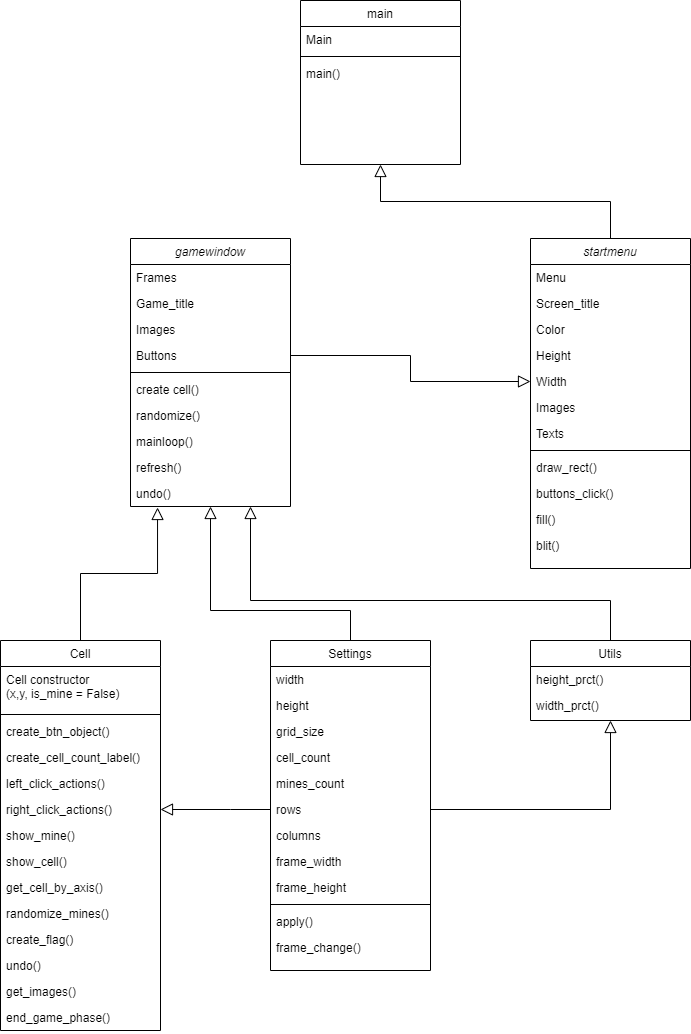
**- Start menu:** when designing the game, start menu is considered after the feeling of lacking features. Start menu adds a new flavor to the bland game. **pygame** library helps in the design of the start menu.

**- Execution file: pyinstaller** converts the whole project folder into one working .exe file. .exe file is already compile to machine code. So, there is no need to install any language interpreter to run the code.

1. ***Class Diagram:***

Below is the class diagram of Minesweeper game made by draw.io.

**Link:** [Minesweeper - diagrams.net](https://app.diagrams.net/#G1jl7mJRS7xQHuWyKMz0ZI9Vb-TQ84ti-T)

****

1. ***References***

***Wikipedia.com:*** [Minesweeper (video game) - Wikipedia](https://en.wikipedia.org/wiki/Minesweeper_(video_game))

***Wikihow.com:*** <https://www.wikihow.com/Play-Minesweeper>

***freeCodeCamp.org***:[Python Game Development Project Using OOP – Minesweeper Tutorial (w/ Tkinter) - YouTube](https://www.youtube.com/watch?v=OqbGRZx4xUc)

***geeksforgeeks.org***:[Creating start Menu in Pygame - GeeksforGeeks](https://www.geeksforgeeks.org/creating-start-menu-in-pygame/)

**-THIS IS THE END OF THE REPORT-**

**THANK YOU FOR READING**