Minesweeper Game

By

Mohd Saif And Bhargav Reddy

Date: 04-07-2022

Introduction

- The objective of Minesweeper is to find and mark all the mines hidden under the gray squares in shortest time possible.
- If the user opens a box containing a mine then user loses the game or if the user identify all the spots without hitting the mines then user wins the game.

Approach

- We have designed a grid with size n × n.
- After first move from user we have added all the mines at random positions by using random library
- We generated numbers in each square box based on the total number of mines in its adjacent square boxes.

Learning

- Packages: random module (import random)

- By using random module we can generate random numbers

- Example : random.randint(3, 9) its generate random number between 3 to 9 including 3 and 9.

Challenges

- After making the first move the user sometimes get landed on one of the mines but this is not the case user have to safely land without hitting any mines.

- To overcome this we first checked the random pairs generated for placing mines is not the first move.

Statistics

- Number of lines of code: 109

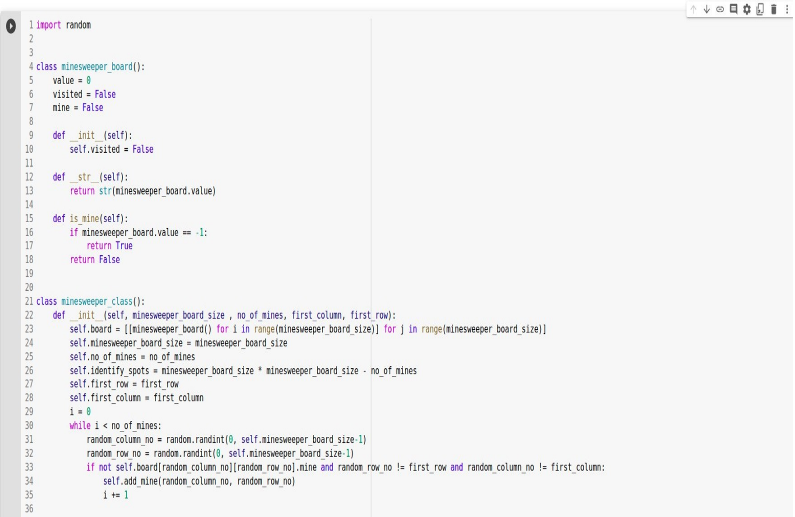
- Number of functions : 6

- Number of class: 2

- Functions Names: play_minesweeper_game(), add_mine(),make_move(),is_winner(),landed_mine(), is_mine().

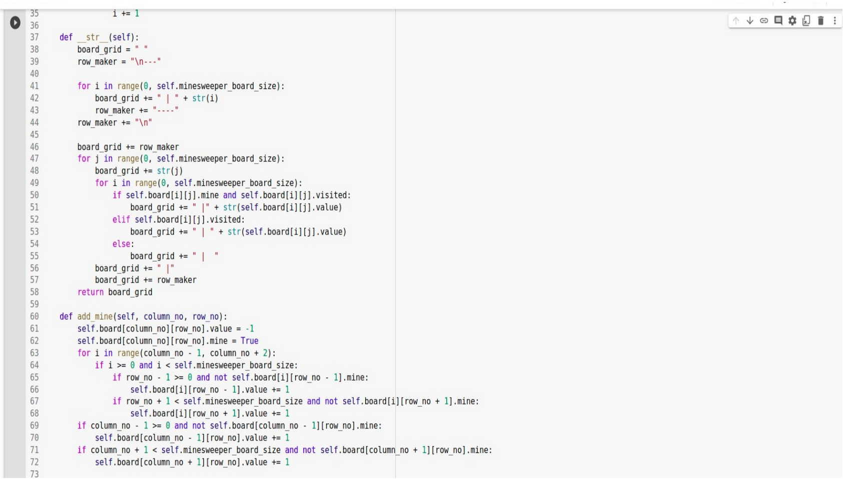
- Class Names: minesweeper_board(), minesweeper_class().

CODE



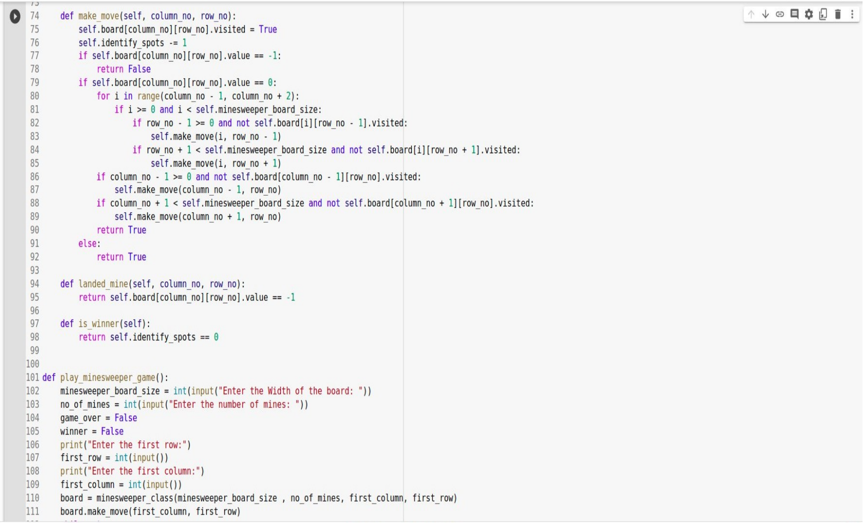
```python
import random


class minesweeper_board():
    value = 0
    visited = False
    mine = False

    def __init__(self):
        self.visited = False

    def __str__(self):
        return str(minesweeper_board.value)

    def is_mine(self):
        if minesweeper_board.value == -1:
            return True
        return False


class minesweeper_class():
    def __init__(self, minesweeper_board_size , no_of_mines, first_column, first_row):
        self.board = [[minesweeper_board() for i in range(minesweeper_board_size)] for j in range(minesweeper_board_size)]
        self.minesweeper_board_size = minesweeper_board_size
        self.no_of_mines = no_of_mines
        self.identify_spots = minesweeper_board_size * minesweeper_board_size - no_of_mines
        self.first_row = first_row
        self.first_column = first_column
        i = 0
        while i < no_of_mines:
            random_column_no = random.randint(0, self.minesweeper_board_size-1)
            random_row_no = random.randint(0, self.minesweeper_board_size-1)
            if not self.board[random_column_no][random_row_no].mine and random_row_no != first_row and random_column_no != first_column:
                self.add_mine(random_column_no, random_row_no)
                i += 1

```

CODE



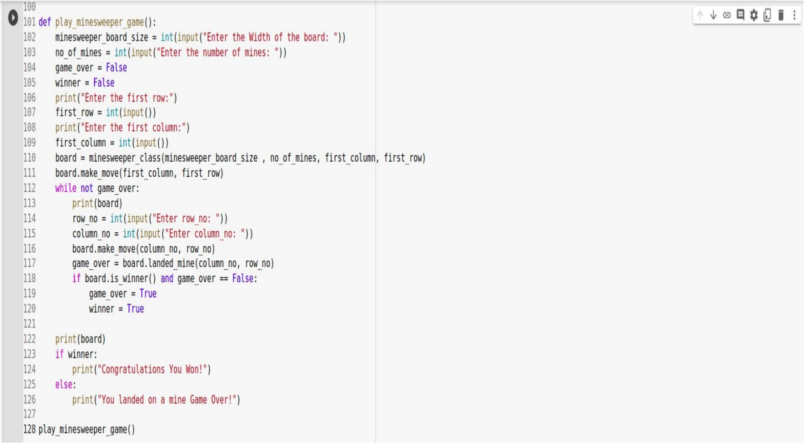
```
                i += 1

    def __str__(self):
        board_grid = " "
        row_maker = "\n---"

        for i in range(0, self.minesweeper_board_size):
            board_grid += " | " + str(i)
            row_maker += "----"
        row_maker += "\n"

        board_grid += row_maker
        for j in range(0, self.minesweeper_board_size):
            board_grid += str(j)
            for i in range(0, self.minesweeper_board_size):
                if self.board[i][j].mine and self.board[i][j].visited:
                    board_grid += " |" + str(self.board[i][j].value)
                elif self.board[i][j].visited:
                    board_grid += " | " + str(self.board[i][j].value)
                else:
                    board_grid += " |  "
            board_grid += " |"
            board_grid += row_maker
        return board_grid

    def add_mine(self, column_no, row_no):
        self.board[column_no][row_no].value = -1
        self.board[column_no][row_no].mine = True
        for i in range(column_no - 1, column_no + 2):
            if i >= 0 and i < self.minesweeper_board_size:
                if row_no - 1 >= 0 and not self.board[i][row_no - 1].mine:
                    self.board[i][row_no - 1].value += 1
                if row_no + 1 < self.minesweeper_board_size and not self.board[i][row_no + 1].mine:
                    self.board[i][row_no + 1].value += 1
        if column_no - 1 >= 0 and not self.board[column_no - 1][row_no].mine:
            self.board[column_no - 1][row_no].value += 1
        if column_no + 1 < self.minesweeper_board_size and not self.board[column_no + 1][row_no].mine:
            self.board[column_no + 1][row_no].value += 1

```

CODE



```
    def make_move(self, column_no, row_no):
        self.board[column_no][row_no].visited = True
        self.identify_spots -= 1
        if self.board[column_no][row_no].value == -1:
            return False
        if self.board[column_no][row_no].value == 0:
            for i in range(column_no - 1, column_no + 2):
                if i >= 0 and i < self.minesweeper_board_size:
                    if row_no - 1 >= 0 and not self.board[i][row_no - 1].visited:
                        self.make_move(i, row_no - 1)
                    if row_no + 1 < self.minesweeper_board_size and not self.board[i][row_no + 1].visited:
                        self.make_move(i, row_no + 1)
            if column_no - 1 >= 0 and not self.board[column_no - 1][row_no].visited:
                self.make_move(column_no - 1, row_no)
            if column_no + 1 < self.minesweeper_board_size and not self.board[column_no + 1][row_no].visited:
                self.make_move(column_no + 1, row_no)
            return True
        else:
            return True

    def landed_mine(self, column_no, row_no):
        return self.board[column_no][row_no].value == -1

    def is_winner(self):
        return self.identify_spots == 0


def play_minesweeper_game():
    minesweeper_board_size = int(input("Enter the Width of the board: "))
    no_of_mines = int(input("Enter the number of mines: "))
    game_over = False
    winner = False
    print("Enter the first row:")
    first_row = int(input())
    print("Enter the first column:")
    first_column = int(input())
    board = minesweeper_class(minesweeper_board_size , no_of_mines, first_column, first_row)
    board.make_move(first_column, first_row)
```

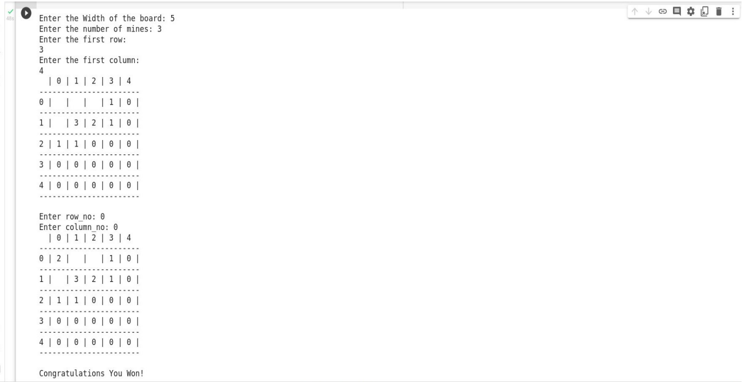
CODE



```python

def play_minesweeper_game():
    minesweeper_board_size = int(input("Enter the Width of the board: "))
    no_of_mines = int(input("Enter the number of mines: "))
    game_over = False
    winner = False
    print("Enter the first row:")
    first_row = int(input())
    print("Enter the first column:")
    first_column = int(input())
    board = minesweeper_class(minesweeper_board_size , no_of_mines, first_column, first_row)
    board.make_move(first_column, first_row)
    while not game_over:
        print(board)
        row_no = int(input("Enter row_no: "))
        column_no = int(input("Enter column_no: "))
        board.make_move(column_no, row_no)
        game_over = board.landed_mine(column_no, row_no)
        if board.is_winner() and game_over == False:
            game_over = True
            winner = True

    print(board)
    if winner:
        print("Congratulations You Won!")
    else:
        print("You landed on a mine Game Over!")

play_minesweeper_game()
```

CODE

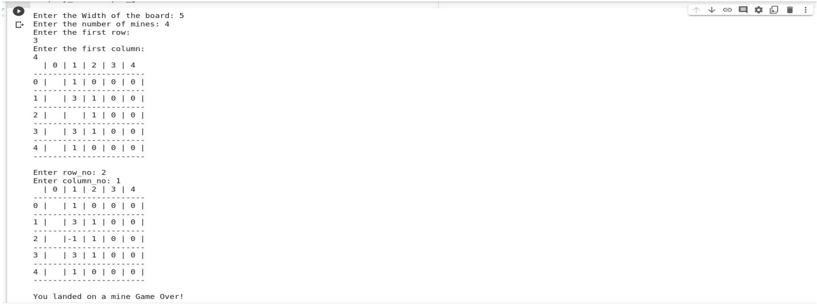
CODE OUTPUT



```
Enter the Width of the board: 5
Enter the number of mines: 3
Enter the first row:
3
Enter the first column:
4
  | 0 | 1 | 2 | 3 | 4
-----------------------
0 |   |   |   | 1 | 0 |
-----------------------
1 |   | 3 | 2 | 1 | 0 |
-----------------------
2 | 1 | 1 | 0 | 0 | 0 |
-----------------------
3 | 0 | 0 | 0 | 0 | 0 |
-----------------------
4 | 0 | 0 | 0 | 0 | 0 |
-----------------------

Enter row_no: 0
Enter column_no: 0
  | 0 | 1 | 2 | 3 | 4
-----------------------
0 | 2 |   |   | 1 | 0 |
-----------------------
1 |   | 3 | 2 | 1 | 0 |
-----------------------
2 | 1 | 1 | 0 | 0 | 0 |
-----------------------
3 | 0 | 0 | 0 | 0 | 0 |
-----------------------
4 | 0 | 0 | 0 | 0 | 0 |
-----------------------

Congratulations You Won!
```

CODE

CODE OUPUT



```
Enter the Width of the board: 5
Enter the number of mines: 4
Enter the first row:
3
Enter the first column:
4
  | 0 | 1 | 2 | 3 | 4
-----------------------
0 |   | 1 | 0 | 0 | 0 |
-----------------------
1 |   | 3 | 1 | 0 | 0 |
-----------------------
2 |   |   | 1 | 0 | 0 |
-----------------------
3 |   | 3 | 1 | 0 | 0 |
-----------------------
4 |   | 1 | 0 | 0 | 0 |
-----------------------

Enter row_no: 2
Enter column_no: 1
  | 0 | 1 | 2 | 3 | 4
-----------------------
0 |   | 1 | 0 | 0 | 0 |
-----------------------
1 |   | 3 | 1 | 0 | 0 |
-----------------------
2 |   |-1 | 1 | 0 | 0 |
-----------------------
3 |   | 3 | 1 | 0 | 0 |
-----------------------
4 |   | 1 | 0 | 0 | 0 |
-----------------------

You landed on a mine Game Over!
```

THANK YOU