PROJECT REPORT

Team Members:

Akshay Patidar - 160050050
Tanmaya Shekhawat - 160050051
Ashutosh Kumar Verma - 160050063

Introduction-

Photo of Start Screen.

We have implemented the game of minesweeper.

Description of The Problem :

- There will be a grid in which some squares will have mines and others will have a number equal to number of mines surrounding that cell.

- Each cell will have two states, opened and closed.
- If the player left-clicks a closed cell
  Case I : If the cell is surrounded by 1 or more mines then cell will show that number.
  Case II : If the cell has a mine, the game is over.
  Case III : If the cell is not surrounded by a mine then all the sorrounding cells will be opened automatically.
- If a player right-clicks on a closed cell,he puts a flag on that cell indicating that cell has a mine.
- If the player identifies all the mines in the grid correctly, he will win.

Design Of The Solution :

We have defined an ADT cell which has the following members.

- **Col :** It will tell the column number of the cell.
- **Row :** It will tell the row number of the cell.
- **IsOpen(True/False):** It will describe the state of a cell.(Initially it will be false for all cells.)
- **NeighbourMineCount:** It will denote the number of mines surrounding a cell. It can have a value between 1 to 3 for corner cells, 1 to 5 for edge cells and 1 to 8 for other cells.
- **HasMine (True/False):** It will tell whether the cell has mine or not.(Initially it will be false for all cells.)
- **HasFlag (True/False):** It will tell whether the cell has flag or not.(Initially it will be false for all cells.)

We have also defined a 2d-vector which will store all the cells of the grid. The 2-d vector has been implemented in the form of many 1-d vectors.
After defining all these when the user will hit play button we will assign a particular number of mines (depends on difficulty level) **randomly** in the cells and set HasMine = true for these cells.

If the player **left-clicks** on a cell and

- **If IsOpen=false** on that cell, then
  set **IsOpen=true**, and
  if **HasMine=true** on that cell then Game Over, otherwise it will show the number of surrounding mines. If there is no mine surrounding it then it will automatically left-click surrounding cells .
  If all the cells which have HasMine=false also have IsOpen=true the player will win.
- **If IsOpen =true** then do nothing.

If the player **right-clicks** a cell

- **If IsOpen=false** set HasFlag=true on that cell.
- **If HasFlag=true** set HasFlag=false on that cell.
- **If IsOpen=true** then do nothing.

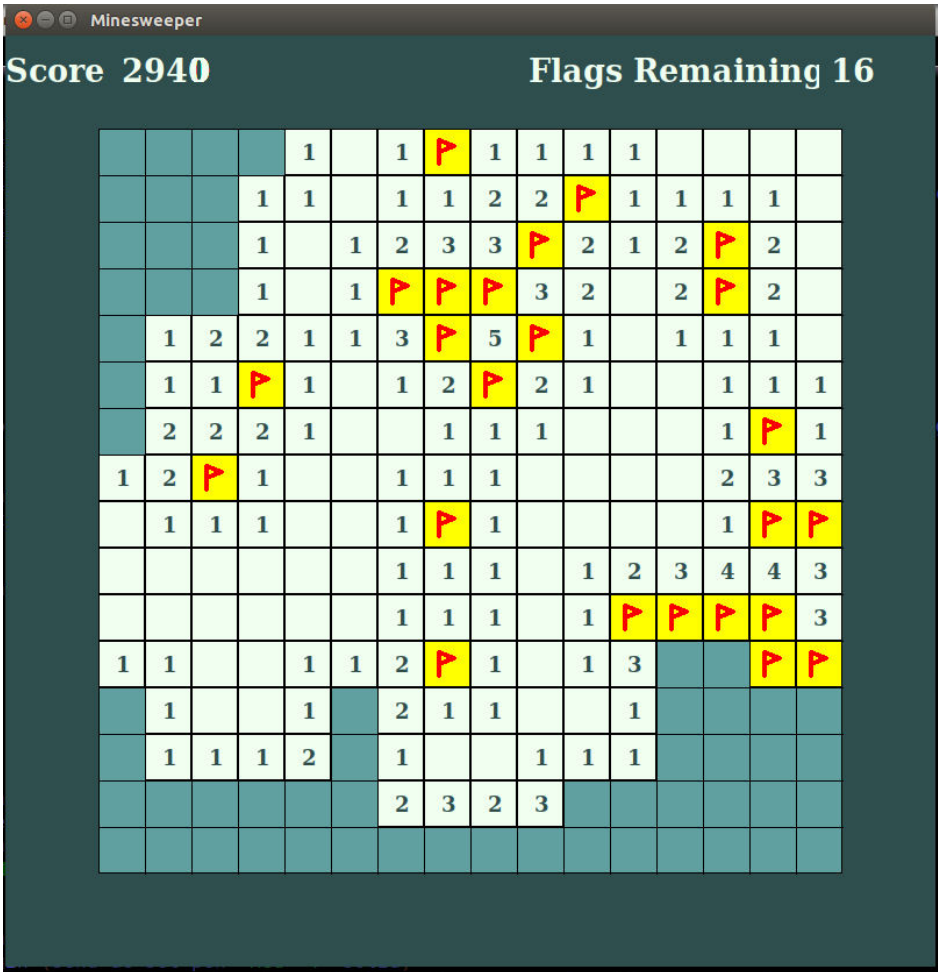

Photo of medium difficulty game.

Sample Input/Output :

The input is given by mouse-clicks. We identify the cell on which the click has been made. Then we perform appropriate operations as defined above. The output screen also shows the number of flags remaining. It prints a suitable message when the game gets over.

Limitations and Bugs :

1. We have not implemnted any timer in the game.
2. On maximising the frame, it does not scale to the full-screen.
3. There are no sound effects in the game.