MINESWEEPER

Final Project Report:

1.Team Details:

Lab day: Wednesday

Lab Batch Number: 3-51

Lab JTA: Subhasmita Mahalik 113050073

Team Members:

1.Ranjeet Kumar Singh	115320013
2.Rishav Goswami	115060023
3.Rohan Rao	115280003
Ruchi Dahiya	115090028
5.Sahil Vaidya	11D170008
6.Sandeep Verma	110040063
7.S.S.Kausik	110050003

2.Introduction:

Minesweeper is a single-player visual game. The object of the game is to clear an abstract minefield without detonating a mine. Minesweeper cannot always be solved with 100% certainty, and may require the occasional use of probability to flag the square most likely to have a mine. In other words, one must sometimes guess to solve a minesweeper puzzle.

3.Modules of the Program:

a. initialize.cpp:

Contains the declaration of the global variables and the initialization of the variables needed to start the game.

b.randomize.cpp

Contains the function definition of fairrandomize which randomly assigns 35 mines in the grid such that 35 unique locations are assigned and also the location of the first mouse click is not assigned a mine.

c.generategrid.cpp

Used to generate the grid numbers according to the mines around them.

d.graphics:
Includes files
displayn.cpp
displaymineflag.cpp
display8square.cpp
display0polygon.cpp
Used to display the graphics(related to EzWindows)[displaying the grid,displaying the flags, displaying the mines and the numbers.]

e.challenge.cpp
Contains the function challenge() used for the Challenge functionality.

You can secure a visible cell using flags. If you correctly place flags on all the mines around a visible cell and secure that place, clicking that cell will open the non-mine cells around that cell.

However beware that if you place flags wrongly on the non-mine cells and use the challenge , you will lose the game for wrongly securing a cell.
If you have placed flags less than the number of mines and click on the cell, nothing will happen.

f.defuse.cpp
Contains the function defuse() which is used to implement the defusal kit functionality.
Owing to technological advances, you can use the defusal kit to detect and defuse the mines. To use the defusal kit and then click on the two cells where you want to defuse the mine. A new window will open, and buttons will displayed after a time interval and you need to press the new button before another button displays.
However you have only 20 seconds and 3 attempts to click the button.
If you are unable to defuse within 20 seconds, or three clicks , you will detonate the mines and lose the game.
Also a time penalty on 10 seconds is added for using the defusal kit.

g.timeclock.cpp
Used to implement the timer

h. newgame.cpp
Used to start a new game of minesweeper

i.saveloadgame.cpp
Used to reload the previous game. Contains the functions reloadgame() and

```
initialize_reload();
```

j.gamestatuscheck.cpp
Contains the function gamestatuscheck() which is used to check whether all the cells have been uncovered or not.

k.highscr.cpp
Contains the code for storing high scores of the game.

l.keyboard.cpp
Used to make a on screen keyboard needed to input the name of the person who has made a high score.

4.Status of the Project:

The project was completed beyond our first stage target which was the basic display of the cells and response to click of the mouse to do the specific action.
The flag mode was added as an addition. Also the project included the challenge and a completely new function-defusal kit --- a new for minesweeper.
Also high scores and on-screen keyboard(a last minute entry) was added to the project.

5.Future Work:

We could add variable rowsize and column size and variable number of mines. Other than that there can be no further addition to minesweeper.

6.Individual Efforts:

Sahil Vaidya, Sandeep Kumar Verma ,Rishav Goswami – contributed to randomize.cpp and overall design.
Ruchi Dahiya – contributed to generategrid.cpp
Rohan Rao - contributed to the algorithm and code of the display zero function.
S.S.Kausik- contributed to the graphics (EzWindows), defuse.cpp,challenge.cpp,newgame.cpp,highscr.cpp,keyboard.cpp
Sahil Vaidya- debugged the challenge function.

7.Acknowlegdements:

We thank Prof. D. B. Phatak for giving us an opportunity to make a CS project which helped us learn various aspects of programming and team management. We thank our Lab JTA for her valuable support. Also we thank Aamod Kore for his help in developing the high scores and onscreen keyboard. Also we thank all our friends who have given useful inputs in various stages of the project. We also thank Saswat Padhi and Nikunj Saunshi for their valuable help in the graphics of EzWindows.