Design Document
Kiana Partovi
Minesweeper Game

Data

Minesweeper Board- 2d array of ints (8x8)
Saved Flags- array of chars
Saved Uncovered cells - array of ints
Size of arrays - integer value
Printing at end - array of character strings

Game Play

The goal of this game is to complete the grid without hitting any bombs throughout gameplay. This game starts out with a blank board. The user first gets to choose which cell, by giving its coordinates, in the 8x8 grid they would like to interact with. They can either mark the cell with a bomb flag, uncover to reveal what is in the cell, or check the surrounding cells. By marking the cell with a bomb flag, the user is noting that that is where they think a bomb. Uncovering what is in the cell will reveal a number. This number will inform the user of how many bombs are potentially touching that current cell. The user has to use a process of elimination to continue to solve the game. When the user checks surrounding cells, they basically give up and the game ends. The user has the option to replay or exit the game.

Functions

main()
Functionality: In a do while loop calls getUserInput function and starts a switch case for the functionality of the entire game based on the three options. In those three options other functions will be called to complete the game play.

readDataFromFileAndStoreInArrays()
Input Parameters: File pointer, 2D array
Returned Output: None
Functionality: This function will scan from the file and store all of the data into the 2d array.

translatingNumbers()
Input Parameters: File pointer, 2D array
Returned Output: None
Functionality: Calling readDataFromFileAndStoreInArrays and then will continue to switch on the numbers. If it is a 99 it will translate to a bomb which is a char X. If it is any other number it will display that integer. If it is a 0, it will translate to a dash.

getUserInput()
Input Parameters: None
Returned Output: integer of users option choice
Functionality: This function will call the displayBoard function and then ask the user ‘which cell?’ After this the user will input the coordinates. We will scan this into an integer array. Then We will display three options. The user will select a number(integer) for the option they want and the function will return this number.

displayBoard()
Input Parameters: char or int 2D array based on if statement, int size of array
Returned Output: None
Functionality: Displays the board with printf statements.

readInData()
Input Parameters: 2d array of character strings
Returned Output: None
Functionality: Stores the numbers in the matrix into a 2d array to really start the program.

markCellWithBombFlag()
Input Parameters: 2d array of board, size of array
Returned Output: None
Functionality: This function will print a character X to the array at whichever point the user chose to mark the cell with a bomb flag.

uncoverCell()
Input Parameters: 2d array of board, size of array
Returned Output: None
Functionality: This cell will need to reveal the adjacencies of the surrounding cells to see if there is a bomb. It will need to call another function that does this check and this other function will include counters.

counterFunction()
Input Parameters: 2d array of board, size of array
Returned Output: None
Functionality: This function will do the counter check to see how many bombs are surrounding.

checkSurroundingCells()
Input Parameters: 2d array of board, size of array
Returned Output: None
Functionality: This function will check the bound to not exceed outside of the 2d array.