Exercise 1

Guidelines

This assignment is for making sure you completed 1181 with all of the requisite knowledge to do well in this class. I am purposely not including everything in these instructions.

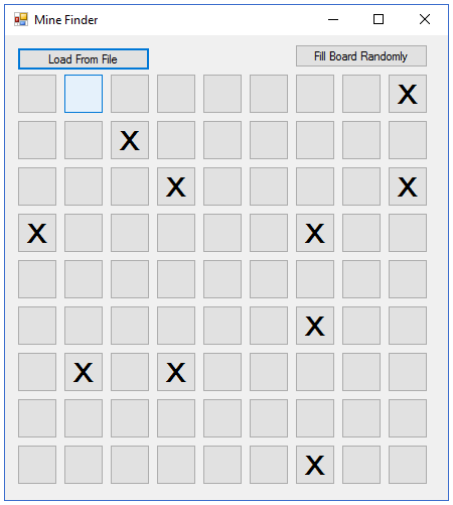
NOTE: This assignment is uses principles we discussed in 1181. If you have trouble solving this solution on your own, it may be wise to consider retaking 1181.

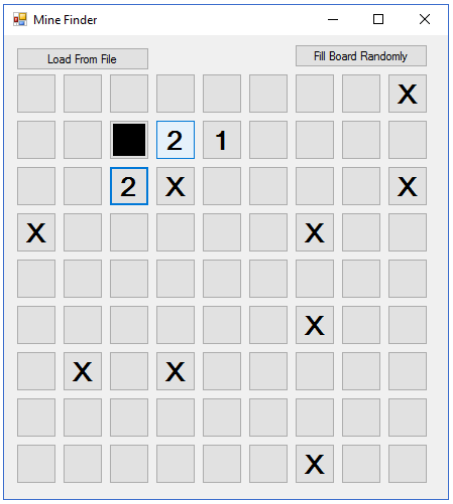
Instructions

You will be making a super simple application that will either read in a text file for a pattern or randomly populates a 9x9 array with bombs that relates to a grid of buttons that a user will click. Each button reacts by either flagging itself or shows a count the adjacent bombs.

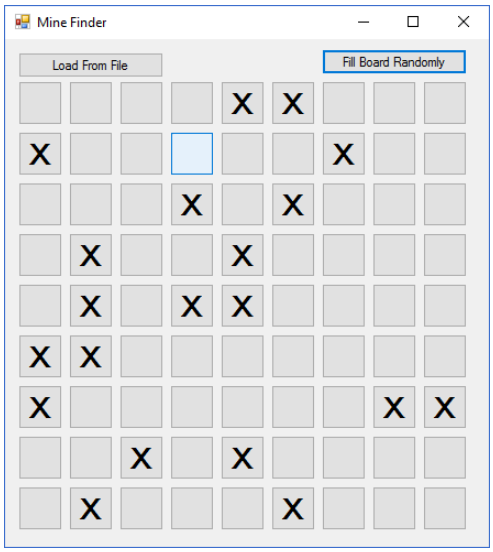
Requirements

- Give the form an appropriate name (frmXxxxx) and title for the application.
- You can modify the layout as needed.
- 81 buttons in a 9x9 grid need to be added to your form. The name of each button should be btn_ROW_COL, where ROW = the index of the row from the top of the grid and COL = the index of the column of the grid from the left. I.E. the top-left most button in the grid should have the name “btn_0_0” and the bottom-right most button should have the name “btn_8_8”.
- Two additional buttons need to exist
    - Load From File
        - This button loads a map from the provided file game.txt into a 2D array that will track where the bombs are on the board. In the file, anywhere there is an “x” there is a bomb.
        - For testing, it might be useful to display to the button if the bomb is there so that you can validate it against your file.
    - Fill Board Randomly
        - Randomly fill the grid with bombs at random locations.
- When a given button is pressed in the grid, the should be changed in the following ways:
    - If the array shows it is a bomb, set the background of the button to a different color (I used black)
    - If it is not a bomb, count how many bombs are it’s immediate neighbors. And set the text on the button to be that count.
    - Make sure you check the array not the buttons.





- You will need to use at least the following methods:
    - FillArrayFromFile
        - Fills the array from the given file. You will need to use a try catch block when handling the file.
    - FillArrayRandomly
        - Fills the array randomly.
    - CountBombs
        - Accepts a column and row index and counts the number of bombs immediately adjacent to the given cell.
        - E.g. the blue cell to the right should have a count of 2.
    - Any more methods needed.



- I have provided two more methods for you to use:
    - The SetButtonClicks method sets the click events for all of your buttons in the grid if you have named them correctly. This should be called in the constructor for the form and after the controls have been initialized.

```
/// <summary>
///  Set the click events for all of the buttons in the grid. Made by Professor Holmes.
/// </summary>
private void SetButtonClicks() {
    btn_0_0.Click += btnBomb_Click; btn_0_1.Click += btnBomb_Click; btn_0_2.Click +=
btnBomb_Click;
btn_0_3.Click += btnBomb_Click; btn_0_4.Click += btnBomb_Click; btn_0_5.Click += btnBomb_Click;
btn_0_6.Click += btnBomb_Click; btn_0_7.Click += btnBomb_Click; btn_0_8.Click += btnBomb_Click;
btn_1_0.Click += btnBomb_Click; btn_1_1.Click += btnBomb_Click; btn_1_2.Click += btnBomb_Click;
btn_1_3.Click += btnBomb_Click; btn_1_4.Click += btnBomb_Click; btn_1_5.Click += btnBomb_Click;
btn_1_6.Click += btnBomb_Click; btn_1_7.Click += btnBomb_Click; btn_1_8.Click += btnBomb_Click;
btn_2_0.Click += btnBomb_Click; btn_2_1.Click += btnBomb_Click; btn_2_2.Click += btnBomb_Click;
btn_2_3.Click += btnBomb_Click; btn_2_4.Click += btnBomb_Click; btn_2_5.Click += btnBomb_Click;
btn_2_6.Click += btnBomb_Click; btn_2_7.Click += btnBomb_Click; btn_2_8.Click += btnBomb_Click;
btn_3_0.Click += btnBomb_Click; btn_3_1.Click += btnBomb_Click; btn_3_2.Click += btnBomb_Click;
btn_3_3.Click += btnBomb_Click; btn_3_4.Click += btnBomb_Click; btn_3_5.Click += btnBomb_Click;
btn_3_6.Click += btnBomb_Click; btn_3_7.Click += btnBomb_Click; btn_3_8.Click += btnBomb_Click;
btn_4_0.Click += btnBomb_Click; btn_4_1.Click += btnBomb_Click; btn_4_2.Click += btnBomb_Click;
btn_4_3.Click += btnBomb_Click; btn_4_4.Click += btnBomb_Click; btn_4_5.Click += btnBomb_Click;
btn_4_6.Click += btnBomb_Click; btn_4_7.Click += btnBomb_Click; btn_4_8.Click += btnBomb_Click;
btn_5_0.Click += btnBomb_Click; btn_5_1.Click += btnBomb_Click; btn_5_2.Click += btnBomb_Click;
btn_5_3.Click += btnBomb_Click; btn_5_4.Click += btnBomb_Click; btn_5_5.Click += btnBomb_Click;
btn_5_6.Click += btnBomb_Click; btn_5_7.Click += btnBomb_Click; btn_5_8.Click += btnBomb_Click;
btn_6_0.Click += btnBomb_Click; btn_6_1.Click += btnBomb_Click; btn_6_2.Click += btnBomb_Click;
btn_6_3.Click += btnBomb_Click; btn_6_4.Click += btnBomb_Click; btn_6_5.Click += btnBomb_Click;
btn_6_6.Click += btnBomb_Click; btn_6_7.Click += btnBomb_Click; btn_6_8.Click += btnBomb_Click;
btn_7_0.Click += btnBomb_Click; btn_7_1.Click += btnBomb_Click; btn_7_2.Click += btnBomb_Click;
btn_7_3.Click += btnBomb_Click; btn_7_4.Click += btnBomb_Click; btn_7_5.Click += btnBomb_Click;
btn_7_6.Click += btnBomb_Click; btn_7_7.Click += btnBomb_Click; btn_7_8.Click += btnBomb_Click;
btn_8_0.Click += btnBomb_Click; btn_8_1.Click += btnBomb_Click; btn_8_2.Click += btnBomb_Click;
btn_8_3.Click += btnBomb_Click; btn_8_4.Click += btnBomb_Click; btn_8_5.Click += btnBomb_Click;
btn_8_6.Click += btnBomb_Click; btn_8_7.Click += btnBomb_Click; btn_8_8.Click += btnBomb_Click;
}
```

    - The btnBomb)Click method handles the button click event for each of the buttons above. I have also helped you by showing you how to get the button that was clicked from the parameters

passed in. You will need to get the row and column from the button name. Make sure to use this row and column set to check your array.

```
/// <summary>
///  Handle the button click for each of the buttons. Started by Professor Holmes.
/// </summary>
/// <param name="sender"> Button sending the event. </param>
/// <param name="e"> Extra variables from the button click event. </param>
private void btnBomb_Click(object sender, EventArgs e) {
    Button btn = (Button)sender;
    // btn.Name is btn_ROW_COL
}
```

- Another line of code you will find useful is the following, which get a button on the form based on its name. This will help you get any button in the grid based on a given row and column.

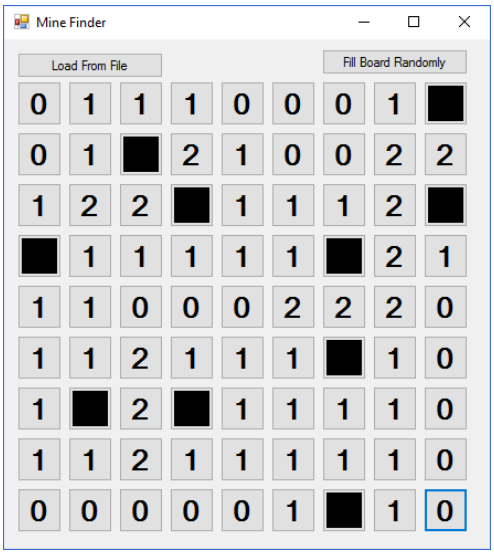
```
Button b = (Button)this.Controls["btn_" + row + "_" + column];
```

Hints:

This should help you visualize counting adjacent cells.

r-1, c-1	r-1, c	r-1, c+1
r, c-1	r, c	r, c+1
r+1, c-1	r+1, c	r+1, c+1

If every button has been pressed when using the file data it should look like this:



File Data:

```
OOOOOOOOX
OOXOOOOOO
OOOXOOOOX
XOOOOOXOO
OOOOOOOOO
OOOOOOXOO
OXOXOOOOO
OOOOOOOOO
OOOOOOXOO
```