MineField 1.0

Time required: 120 minutes

Comment each line of code.

Create a GUI Java program that plays MineField against the computer. The user attempts to click 10 panels before hitting the bomb. This project will setup the GUI, the gameplay will be added in the next version.

Create a jGrasp project named MineField 1.0 and a class named MineField.

Requirements

1. Add the following import statements:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.Random;
```

2. Place the following code at the top of the program to create a random number pool for the mine.
    - Create constants for the number of rows, cols, and gap for the BorderLayout.
    - Create a constant **NUM** that gives the number of panels based on how mand rows and cols are specified.
    - Create an array of JPanels

```
// Create an array of JPanel's for the grid
private JPanel[] panel = new JPanel[NUM];
```

3. Build the following menus:
    - File, Exit
    - Options, New Game, Radio Buttons: Easy: 5 squares to win, Medium: 10 squares to win, Difficult: 15 squares to win.
    - Help, About
4. Use a BorderLayout for the main program.
    - Create and add a JPanel called **gameBoard** to the BorderLayout.CENTER of the main application.
    - Create the gameboard pieces from JPanels using a GridLayout.

- Hint: Use a For Next loop go through the array, create the JPanels, and add them to the gameBoard.
- Create a custom color: **private Color cougarBlue = new Color(0,58,112);**
- Use the custom color for the gameboard panels: **panel[x].setBackground(cougarBlue);**

- Create a statusBar from a JPanel using FlowLayout.LEFT.
- Add the statusBar to the BorderLayout.SOUTH.

Submission

1. Test your finished project. Make corrections as necessary.
2. Zip up the project folder and submit it to Blackboard.

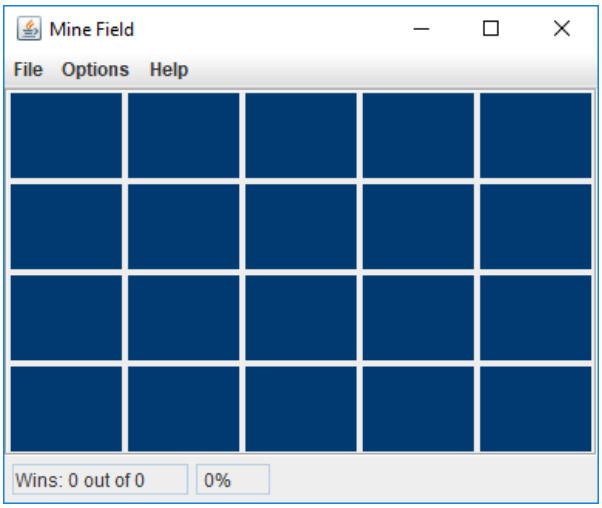

Figure 1 – Sample application design