CS102- Algorithms and Programming II

Lab 05

Lab Objectives: Basic GUI Applications

Notes:

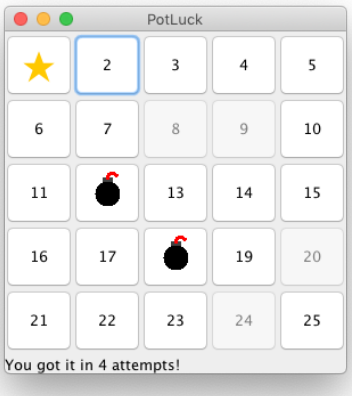
- For all labs in CS 102, your solutions must conform to these CS101/102 style guidelines.
- Create a Java Project named Lab05. Put all of your classes in this project.
- Remember to include **javadoc comments** for each class and method.
- Upload your solution **as a single .zip file** to the Lab05 assignment by the end of your section's lab session in the week of March 29. You must use the following naming convention: Lab05_Surname_FirstName.zip where Surname is your family name and FirstName is your first name. You may upload multiple times; the last upload will be considered.

Introduction

Use VS Code or your favorite Java IDE, but write the code by hand. You will also need the Java documentation, the textbook, and maybe the course slides. It will take time, but doing it this way should help you really understand it, and so enable you to build bigger and better systems in the future. Don't just solve the problem one way, try to think what alternative solutions there are, and what the advantages and disadvantages of each might be. Do each part in its own project within the lab05 workspace.

For the following, please try to use Java's Swing components to build the user-interface, basing your design on a JPanel, so that you can easily add one or more instances of it to a JFrame or another JPanel.

a. **Pot Luck**
   Design and implement a GUI application that presents a game based on a 5 by 5 matrix of buttons. One of the buttons (selected at random) "hides" the prize, while two of the buttons (selected at random, should be different from each other and prize button) hide bombs. A status bar at the bottom of the window shows the number of guesses. When the prize button is pressed, the status bar shows "You got it in x attempts!". When one of the bomb buttons is pressed, the status bar shows "Sorry! You are blown up at attempt x!" Which buttons hid the prize and which ones hid bombs should be revealed to the user. As shown in the screen capture below.

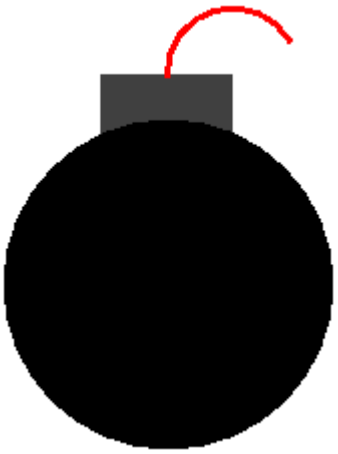


There are three kinds of buttons: regular buttons, bomb buttons, and a prize button. Implement the latter two as subclasses of JButton. Initially, all types of buttons show their text (button number in this case). Disable the buttons that the user clicked, so that they cannot be clicked again. Java will gray out these buttons (see buttons 8, 9, 20, and 24 in the screenshot above). At the end of the game the prize button and bomb buttons should reveal their content.

In order to display the orange star as the prize, change button text to Unicode Black Star ★ (9733 or U+2605). First, try if your editor supports Unicode characters in the source code by copying and pasting the star into your code. If not, you can use the Unicode escape sequence "\u2605" instead. Increase the font size to a value that is proportional to the size of the button. Set the text color (foreground) to orange. You may also need to call `setOpaque(true)` for color change to take effect. Because the button is grayed out when it is disabled, the color will not show when the user clicks the prize button. Therefore, the prize button should not be disabled; or you can enable it back.

Bomb button should draw the bomb by overriding `paintComponent()`. There are two very important considerations when you are overriding `paintComponent()`. First, we should explicitly let our superclass draw itself by calling its `paintComponent()` method. Otherwise, the result will not be what we want. Secondly, whenever the object state changes, e.g., from displaying the button number to revealing the bomb, we should let the Swing framework know that we should be redrawn (Which method should we call?). Try omitting these two considerations and see how the application behaves in order to make yourself familiar with potential issues that you may encounter in future.

The image of the bomb consists of three parts: body, cap, and fuse. Body is a black circle. Cap is a gray rectangle. Fuse is a red arc. By default, the arc that you draw will be very thin. Increase the line width, called *stroke*, using Graphics2D methods. Make sure that you set the position and size of your drawing proportional to the button size, and that the drawing is resized when the window is resized (buttons should resize with the window). The detailed bomb should look similar to the following image:



b. Convert Numbers

Design and implement a GUI application that converts decimal, hex and binary numbers, as shown in the figure below. When you enter a decimal value in the decimal-value text field and press the *Enter* key, its corresponding hex and binary numbers are displayed in the other two text fields. Similarly, you can enter values in the other fields and convert them accordingly. You should implement your own methods to do conversions.