

1 Overview

For your seventh lab, you will update your lab 6 program to plays a more interesting game of craps. Specifically this lab covers,

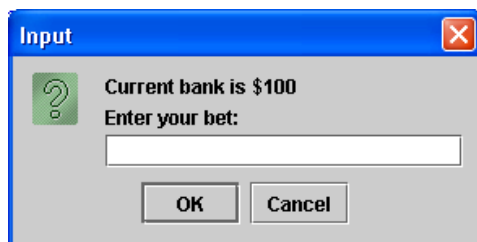
1. Nested selection and repetition structures
2. Methods
3. GUI programming for input and output

Create a new directory called Lab7 off of your C:\userdata\CSC111 directory to save your work.

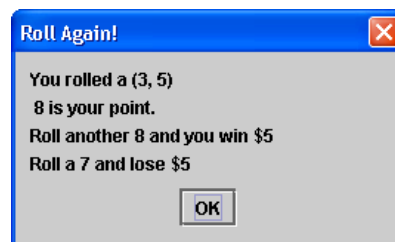
1.1 Betting Java Craps

Create a new file called `BetCraps.java` in your Lab7 directory. Your program will play craps using the rules described in lab 6 and allow the user to bet per game (this is the *improvement*). The game will assume the **user starts with \$100**, called the *bank*. If the user wins the game the bet amount is added to the bank; otherwise, the bet amount is subtracted from the bank. The program will continue to play until the user has a zero bank or decides to quit. As done in lab 6, all input and output must be performed using dialog windows (`JOptionPane`); however, **all input and output must be done using methods**.

The program should start by displaying a welcome message in a dialog window indicating the programmer's name, as done in lab 6. Once the user clicks on `OK` the game should ask the user to enter a bet, as seen in figure 1(a). Obtaining the bet amount **must be done in a method** called `getBet`. This method will prompt the user for a bet (using a dialog window) then return the integer bet amount (via the function name). You can assume the user will enter a number; however, your method **must error check** the input. Therefore re-prompt until a valid number (between 0 and the current bank) is entered. Once a valid bet is entered, the program will simulate rolling the dice. On every roll, your program must display a new dialog window (using a method) indicating the roll and the status of the game. For example, if the user rolls a 7 or 11 on the first roll, the dialog window should tell the user they have won and display the updated bank. If the user rolls a 2, 3, or 12 on the first roll, the dialog window should tell the user they have lost and display the updated bank. Otherwise the the dialog window should display the point and tell the user to roll again, as seen in figure 1(b). The game will continue as described by the rules until the user wins or loses. Once the game is complete, the program will ask to the user if they wish to play again (assuming they have money) as seen in figure 2(b). If the user wishes to play again, a new bet is taken (call your `getBet` method again) and the game repeats. If the user has no money or decides to quit, the program should end.

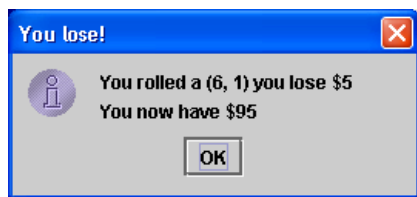


(a) Prompt the user for a bet.

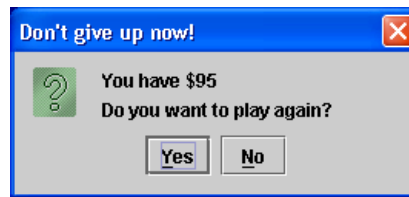


(b) User rolls an 8 on the first roll and must roll again. The point is displayed and instructions on how to win or lose.

Figure 1: Example dialog windows for the betting craps game.



(a) Player rolls a 7 and loses, since 8 was the point. Bet and updated bank is displayed.



(b) Player enthusiastically asked to play again.

Figure 2: Example dialog windows for the betting craps game.

2 GUI Programming

We have used dialog windows for basic GUI input and output since lab 3. The dialog windows are the same for this lab except when the user is asked to play the game again, as seen in figure 2(b). In this case, the program will respond to the button clicked instead of text collected in a dialog box.

Collecting user input through buttons can be done using a `JOptionPane.showConfirmDialog` window. The window will display a message, a `Yes` button, and a `No` button. If the user clicks on the `Yes` button the function will return zero; otherwise, if the user clicks on the `No` button the function returns one. For example, the following code segment will display a dialog window and store the result in the `int` variable `choice`.

```
int choice; // 0 for Yes and 1 for No
choice = JOptionPane.showConfirmDialog(null, "Question", "Window Title",
                                       JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
```

3 Turn-In Checklist

Exactly what do I turn-in to receive credit for this lab? All code **must** adhere to the style guidelines specified for this course. Submit the following electronically **and** give print-outs to your TA

- Your betting craps program called `BetCraps.java`
 - Input and output must be done using methods.
 - Test your program to be certain it works correctly.
 - Make certain the output is as described in the assignment.

All work is due at the end of your lab!
No late work will be accepted!