Craps

Allows you to play a game of craps via a GUI.

- **Author:** Rudd Fawcett
- **Course:** Period 1 AP CS, Dr. Miles
- **Files:**
  - `CrapsGame.java`
  - `Die.java`
  - `DieTest.java`
  - `RollingDie.java`
- **Due:** 10/20/2015

CrapsGame.java

The source code **added** for `CrapsGame.java`.

```java
/**
 *  Calculates the result of the next dice roll in the Craps game.
 *  The parameter total is the sum of dots on two dice.
 *  point is set to the saved total, if the game continues,
 *  or 0, if the game has ended.
 *  Returns 1 if player won, -1 if player lost,
 *  0 if player continues rolling.
 */
public int processRoll(int total) {
    int result;
    if (this.point == 0) {
        if (total == 7 || total == 11) {
            result = 1;
        }
        else {
            if (total == 2 || total == 3 || total == 12) {
                result = -1;
            }
            else {
                result = 0;
                this.point = total;
            }
        }
    }
    else if (total == this.point) {
        result = 1;
        this.point = 0;
    }
    else if (total == 7) {
        result = -1;
        this.point = 0;
    }
    else {
        result = 0;
    }

    return result;
}
```

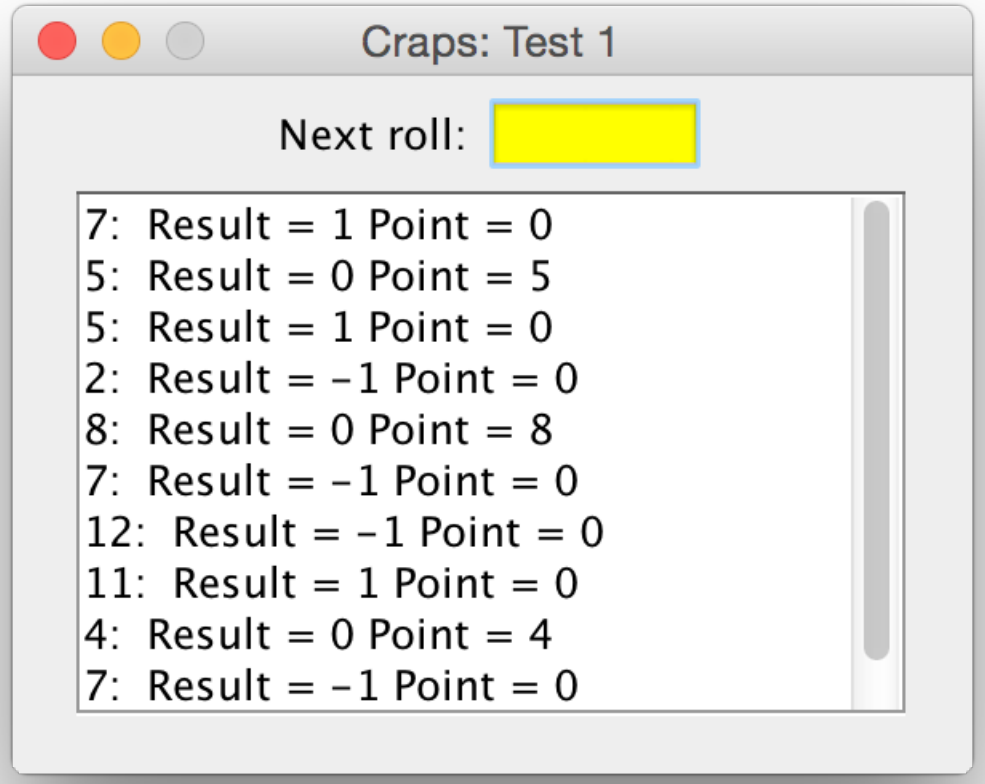
RollingDie.java

The source code **added** for `RollingDie.java`

```
    /**
     * Creates circles based on the number of dots using the "coordinate"
     * like system calculated above, with `x1`, `x2`, `x3`, `y1`, etc.
     */
    switch (numDots) {
      case 1:
        g.fillOval(x2, y2, dotSize, dotSize);
        break;
      case 2:
        g.fillOval(x1, y1, dotSize, dotSize);
        g.fillOval(x3, y3, dotSize, dotSize);
        break;
      case 3:
        g.fillOval(x1, y1, dotSize, dotSize);
        g.fillOval(x2, y2, dotSize, dotSize);
        g.fillOval(x3, y3, dotSize, dotSize);
        break;
      case 4:
        g.fillOval(x1, y1, dotSize, dotSize);
        g.fillOval(x1, y3, dotSize, dotSize);
        g.fillOval(x3, y1, dotSize, dotSize);
        g.fillOval(x3, y3, dotSize, dotSize);
        break;
      case 5:
        g.fillOval(x1, y1, dotSize, dotSize);
        g.fillOval(x1, y3, dotSize, dotSize);
        g.fillOval(x2, y2, dotSize, dotSize);
        g.fillOval(x3, y1, dotSize, dotSize);
        g.fillOval(x3, y3, dotSize, dotSize);
        break;
      case 6:
        g.fillOval(x1, y1, dotSize, dotSize);
        g.fillOval(x1, y2, dotSize, dotSize);
        g.fillOval(x1, y3, dotSize, dotSize);
        g.fillOval(x3, y1, dotSize, dotSize);
        g.fillOval(x3, y2, dotSize, dotSize);
        g.fillOval(x3, y3, dotSize, dotSize);
        break;
    }
```

java CrapsTest1

The result of running `java CrapsTest1` and testing various rolls to test the game functionality.



DieTest.java

The source code for `DieTest.java`.

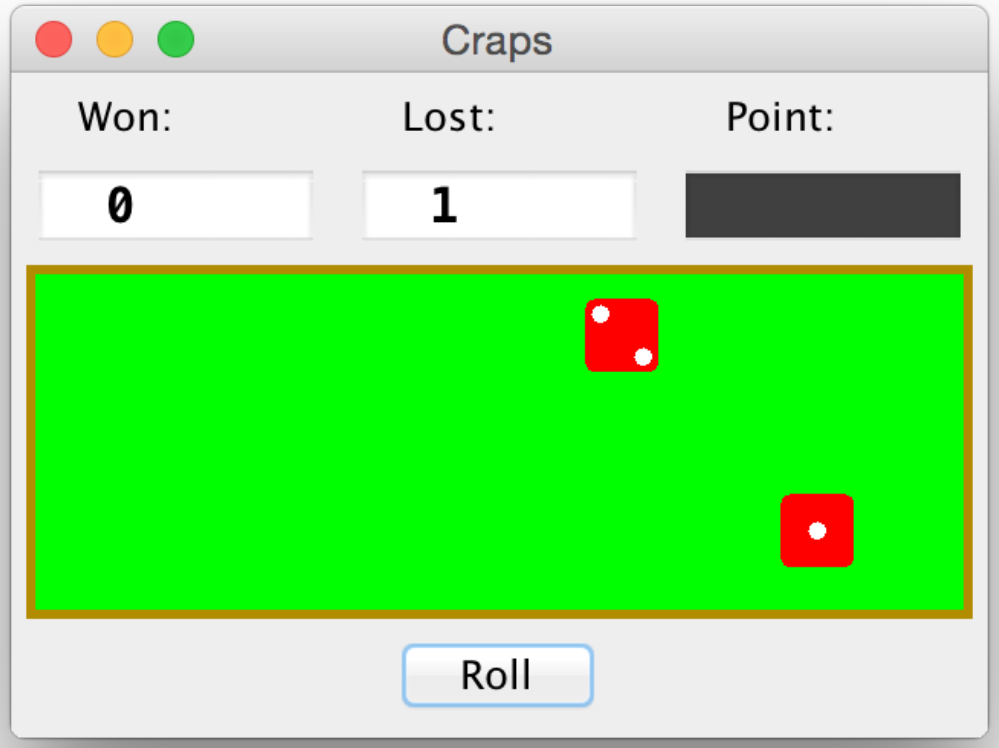
```java
import java.util.Scanner;

public class DieTest {
  public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    System.out.print("How many times would you like to roll? ");

    int times = keyboard.nextInt();

    testRoll(times);
  }

  private static void testRoll(int times) {
    Die die = new Die();
    for(int i = 1; i <= times; i++) {
      die.roll();
      System.out.println("Roll #" + i + ":\t" + die.getNumDots());
    }
  }
}
```

java DieTest

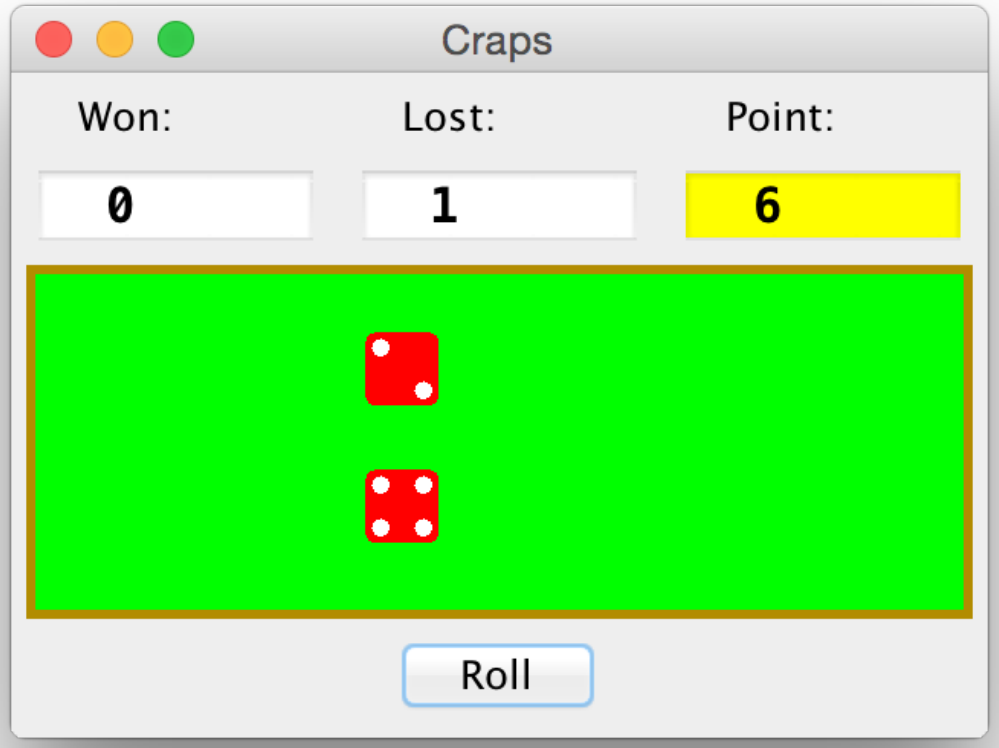
An example of running `DieTest.java` -- you can choose how many rolls to test for.

java Craps

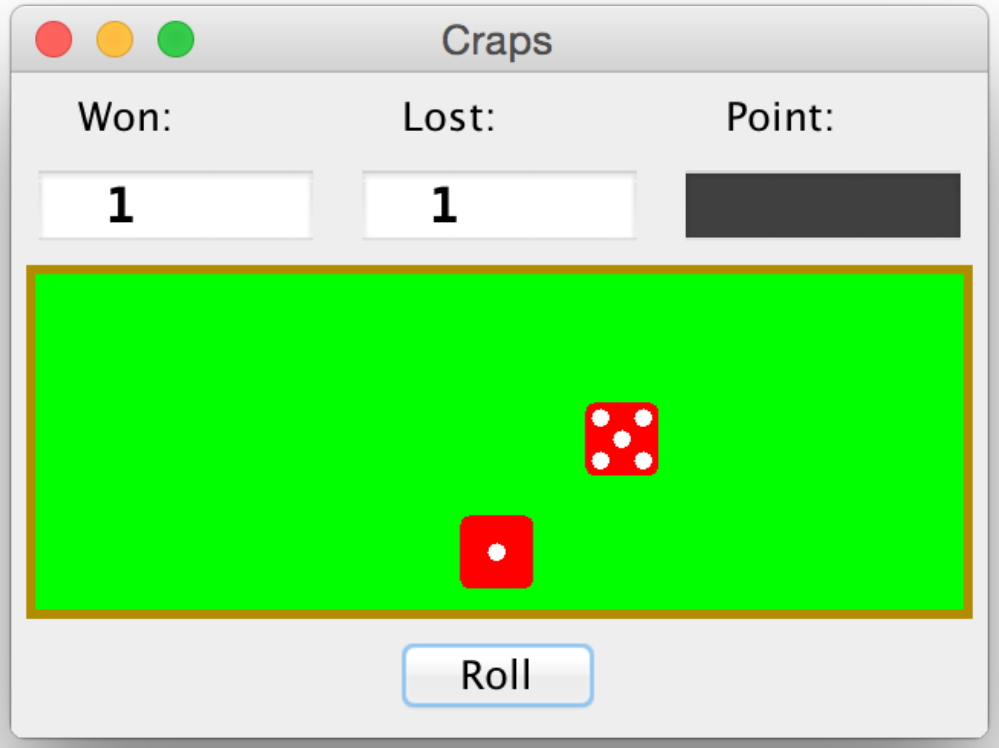
The result of playing the Craps game.



I lost the first roll here because I rolled a 3.



I didn't lose or win here because I rolled a 6.



I won here because I rolled the correct number of points on a subsequent roll.