

# Coding High Roller

This game has 2 players. The one who rolls the higher dice wins.

```
import java.awt.*; import java.awt.event.*; import javax.swing.*; import java.applet.*;

public class HighRoller extends Applet implements ActionListener
{
    JButton bmine, next, bcomp;
    JLabel action, score, pic, pic2;
    int dice1 = 0, dice2=0, cscore=0, uscore=0;

    public void init ()
    {
        JLabel title = new JLabel ("High Roller");
        title.setFont (new Font ("Arial", Font.BOLD, 40));
        title.setForeground (Color.red);

        JLabel words = new JLabel ("You roll. The computer rolls. ");
        JLabel morewords = new JLabel ("Higher roll wins.");
        JLabel mine = new JLabel ("Your dice");
        mine.setFont (new Font ("Arial", Font.BOLD, 18));
        JLabel comp = new JLabel ("Comp dice");
        comp.setFont (new Font ("Arial", Font.BOLD, 18));
        pic = new JLabel (createImageIcon ("blank.gif"));
        pic2 = new JLabel (createImageIcon ("blank.gif"));

        bmine = new JButton ("ROLL Mine");
        bmine.setActionCommand ("1");
        bmine.addActionListener (this);

        bcomp = new JButton ("ROLL Comp");
        bcomp.setActionCommand ("2");
        bcomp.addActionListener (this);
        bcomp.setEnabled (false);

        next = new JButton ("Play again.");
        next.setActionCommand ("3");
        next.addActionListener (this);
        next.setEnabled (false);

        action = new JLabel ("Roll your dice to begin.");
        action.setFont (new Font ("Arial", Font.BOLD, 18));
        score = new JLabel ("Score: You _____, Computer _____");

        add (title);
        add (words);
        add (morewords);

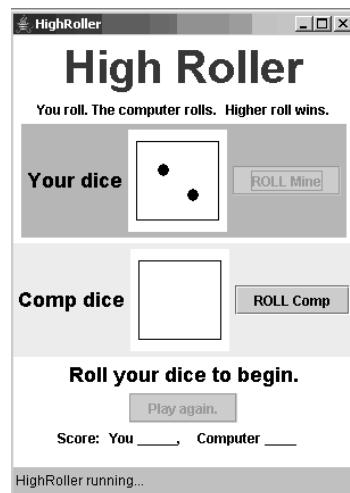
        Panel p = new Panel ();
        p.add (mine);
        p.add (pic);
        p.add (bmine);
        p.setBackground (Color.green);
        add (p);

        Panel p2 = new Panel ();
        p2.add (comp);
        p2.add (pic2);
        p2.add (bcomp);
        p2.setBackground (Color.yellow);

        add (p2);
        add (action);
        add (next);
        add (score);
        resize (300, 380);
    }


    protected static ImageIcon
    createImageIcon (String path)
    //pretend it is here.

```

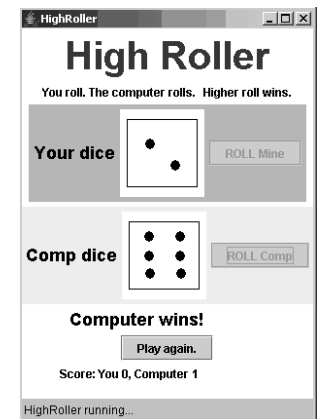


```
public void actionPerformed (ActionEvent e)
{
    if (e.getActionCommand ().equals ("1"))
    {
        }
    else if (e.getActionCommand ().equals ("2"))
    {
        if (dice1 == dice2)

        else if (dice1 > dice2)
        {
        }
        else
        {
        }
    }
    else // e.getActionCommand ().equals ("3")
    {
    }
}
}
```

1.3  T

Name: \_\_\_\_\_



This is the code to play a logic matching game.

```
import javax.swing.*; import java.applet.Applet; import
java.awt.*; import java.awt.event.*;

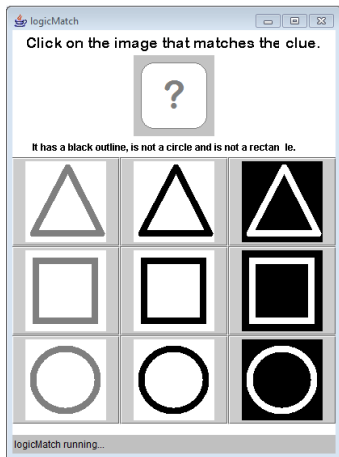
public class logicMatch extends Applet implements ActionListener
{
    int num1 = 1;
    int num2 = 2;
    JLabel clue, message, pic;
    public void init ()
    {
        message = new JLabel ("Click on the image that matches the
        clue.");
        message.setFont (new Font ("Arial Rounded MT Bold",
        Font.PLAIN, 18));
        pic = new JLabel (createImageIcon ("blank.png"));
        clue = new JLabel ("It has a black outline, is not a
        circle and is not a rectangle.");

        Panel g = new Panel (new GridLayout (3, 3));
        JButton a = new JButton (createImageIcon ("11.png"));
        a.addActionListener (this);
        a.setActionCommand ("11");
        g.add (a);
        JButton b = new JButton (createImageIcon ("12.png"));
        b.addActionListener (this);
        b.setActionCommand ("12");
        g.add (b);
        JButton c = new JButton (createImageIcon ("13.png"));
        c.addActionListener (this);
        c.setActionCommand ("13");
        g.add (c);

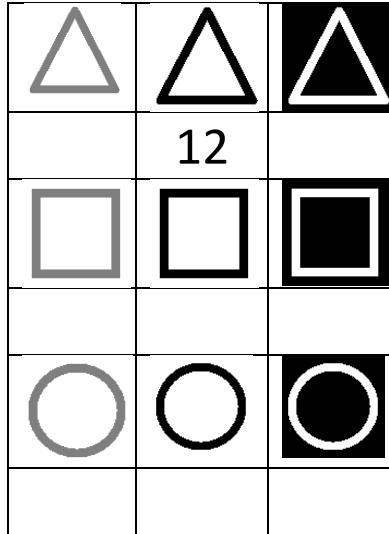
        JButton d = new JButton (createImageIcon ("21.png"));
        d.addActionListener (this);
        d.setActionCommand ("21");
        g.add (d);
        JButton e = new JButton (createImageIcon ("22.png"));
        e.addActionListener (this);
        e.setActionCommand ("22");
        g.add (e);
        JButton f = new JButton (createImageIcon ("23.png"));
        f.addActionListener (this);
        f.setActionCommand ("23");
        g.add (f);

        JButton h = new JButton (createImageIcon ("31.png"));
        h.addActionListener (this);
        h.setActionCommand ("31");
        g.add (h);
        JButton i = new JButton (createImageIcon ("32.png"));
        i.addActionListener
        (this);
        i.setActionCommand
        ("32");
        g.add (i);
        JButton j = new
        JButton
        (createImageIcon
        ("33.png"));
        j.addActionListener
        (this);
        j.setActionCommand
        ("33");
        g.add (j);

        add (message);
        add (pic);
        add (clue);
        add (g);
        add (g);
        resize (400, 500);
    }
}
```



Based on the init method, what are the action commands of each of these pictures? One has been filled in for you.



How are the pictures named?

- a) All of the circles have....
- b) All of the triangles have....
- c) All of the squares have...
- d) All of the grey shapes have....
- e) All of the black shapes have ....
- f) All of the white shapes on black have ...

A Sample Clue:



It has a black outline, is not a triangle and is not a rectangle.

Fill in the code in the actionPerformed method of the logic matching game.

```
public void actionPerformed (ActionEvent e)
{
    int i = Integer.parseInt (e.getActionCommand ());
    //show the user's choice (1 line)

    //check if they are right (1 boolean expression)

    if (
        )
    { //tell them they are right (1 line)

        num1 = (int) (Math.random () * 3 + 1);
        num2 = (int) (Math.random () * 3 + 1);
        //print the new clue (12 lines)

    }
    else //tell them they are wrong (1 line)

}
//end actionPerformed
```

