```java
package example;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

public class RaceGame implements ActionListener {
        // Interface Components
        JButton die1;
        JButton die2;
        JButton roll_Button;
        JButton rightDye;
        JButton right2ndDye;
        JLabel[] cell;
        JPanel raceTrack;
        JLabel mesg;

        boolean roll = true;
        int player1=0; // player1's position
        int player2=14; // player2's position

        int win =0; // no one has won yet
        int clicks =0; // no clicks have occurred yet
        int turnID = 1; // starts with it being player 1's turn

        /**
         * Create the GUI and show it. Invoked from the event-dispatching
thread for
         * thread safety.
         */
        private static void createAndShowGUI() {

                // Create and set up the window.
                JFrame frame = new JFrame("Race Game");
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                Container cont = frame.getContentPane();

                RaceGame app = new RaceGame();
                app.init(cont);
```

(handwritten annotations)

Scott Palmer
Mike

does not have good naming convention

magic #'s

```java
            // Display the window.
            frame.pack();
            frame.setVisible(true);
    }

    public void init(Container cont) {

            die1 = new JButton("?");  // number needs to be determined
            // by a roll

            die1.addActionListener(this);
            die2 = new JButton("?");  // number needs to be determined
            // by a roll

            die2.addActionListener(this);

            roll_Button = new JButton("Roll");
            roll_Button.addActionListener(this);

            rightDye = new JButton("-");  // Number already used
            rightDye.addActionListener(this);
            right2ndDye = new JButton("-");  // Number not available
            right2ndDye.addActionListener(this);

            cont.setLayout(new BorderLayout());
            JPanel buttonRow = new JPanel();
            buttonRow.setLayout(new GridLayout(1, 0));

            // Add the buttons to form the player command row
            buttonRow.add(die1);
            buttonRow.add(die2);
            buttonRow.add(roll_Button);
            buttonRow.add(rightDye);
            buttonRow.add(right2ndDye);

            raceTrack = new JPanel(new GridLayout(3,5));

            raceTrack.setPreferredSize(new Dimension(320,175));
            raceTrack.setBackground(Color.LIGHT_GRAY);
            cell = new JLabel[15];
            for (int i=0; i<15; i++) {
                    cell[i] = new JLabel(i+":", JLabel.CENTER);

cell[i].setBorder(BorderFactory.createLineBorder(Color.black));
                    raceTrack.add(cell[i]);
            }
            cell[0].setText("0:A");  // set initial position of player1
            cell[14].setText("14:B");  // set initial position of
```

Handwritten annotations:

- Not all the same char — should be a var
- by a roll
- by a roll
- Not in the same conformity as other components. Excessive

player2
```
            cont.add(raceTrack, BorderLayout.CENTER);
            cont.add(buttonRow, BorderLayout.SOUTH);

            mesg = new JLabel("Player 1, Press Roll To Start Race
Game");
            cont.add(mesg, BorderLayout.NORTH);
    }

    public static void main(String[] args) {
            // Schedule App's GUI create & show for event-dispatching
thread
            javax.swing.SwingUtilities.invokeLater(new Runnable() {
                public void run() {
                        createAndShowGUI();
                }
            });
    }

    public void actionPerformed(ActionEvent ae) {
            // Find which button was pressed
            JButton b = (JButton) ae.getSource();

            if (b.getText().contains("?") || b.getText().contains("-"))
                    return; // a number button which does not hold a
    valid number

            if (b == roll_Button) { // roll button pressed
                    if (roll == true) { // create new roll
                            if (turnID == 1) { // hard coded numbers -
    change to use random
                                    die1.setText("1");
                                    die2.setText("4");
                                    mesg.setText("Player 1, click a
    number on the left");
                                    roll=false;
                            } else { // hard coded numbers - change to
    use random
                                    rightDye.setText("4");
                                    right2ndDye.setText("1");
                                    mesg.setText("Player 2, click a
    number on the right");
                                    roll=false;
                            }
                    }
                    return;
```

needs to be method for a valid number (handwritten annotation)

use random (handwritten annotation)

```java
        }

        int n;
        // Die buttons for player 1
        if (b== die1) { // use number of this die for move
                n = Integer.valueOf(die1.getText());
                die1.setText("-"); // number no longer available
                clicks++;
                // move player and update the board
                cell[player1].setText(player1+":");
                player1+=n;
                if (player1>14) player1 = 14;
                cell[player1].setText(player1+":A");
        }

        if (b== die2) {
                n = Integer.valueOf(die2.getText());
                die2.setText("-");
                clicks++;
                // update current position to remove player token
                cell[player1].setText(player1+":");
                player1+=n; // player position is updated
                if (player1 > 14) player1 = 14;
                cell[player1].setText(player1+":A");
        }

        // Die buttons for player2
        if (b== rightDye) { // number selected comes from right dye
                n = Integer.valueOf(rightDye.getText());
                rightDye.setText("-"); // mark number as used
                clicks++;
                // remove token from old position and move to new

                cell[player2].setText(player2+":");
                if (player2 > 0) player2-=n;
                cell[player2].setText(player2+":B");
        }

        if (b== right2ndDye) {
                n = Integer.valueOf(right2ndDye.getText());
                right2ndDye.setText("-");
                clicks++;
                cell[player2].setText(player2+":");
                if (player2 > 0) player2-=n;
                cell[player2].setText(player2+":B");
        }
```

Handwritten annotations:

Can be combined (brace grouping die1 and die2 blocks)

position

Can be combined (brace grouping rightDye and right2ndDye blocks)

```java
            // Determine if anyone has won yet
            if (player1>=14) win = 1;
            else if (player2<=0) win = 2;

            // Determine if time to roll again
            if (clicks ==2) { // after 2 clicks time to roll again
                    if (turnID == 1) {
                            turnID = 2;
                            roll = true;
                            rightDye.setText("?");
                            right2ndDye.setText("?");
                    } else {
                            turnID = 1;
                            roll = true;
                            die1.setText("?");
                            die2.setText("?");
                    }

                    clicks = 0;
                    if (win == 0)
                            mesg.setText("Player " + turnID + ", Time
    to hit Roll again");
                    else if (win == 1)
                            mesg.setText("Player 1 Wins!");
                    else
                            mesg.setText("player 2 Wins!");
            }

        } // end of actionPerformed method
    }
```