

Name: Dan Cassidy

Class: CSCI-C 490, Mobile Application Development

Assignment: Homework 6 Part 1

Date: 2015-07-24

```

1  /*-----
2  * Author:      Dan Cassidy
3  * Date:        2015-07-24
4  * Assignment:  HW6-1
5  * Source File: MainActivity.java
6  * Language:    Java
7  * Course:      CSCI-C 490, Android Programming, MoWe 08:00
8  -----*/
9  package dancassidy.bmicalculator;
10
11  import android.app.Activity;
12  import android.os.Bundle;
13  import android.text.Editable;
14  import android.text.TextWatcher;
15  import android.widget.EditText;
16  import android.widget.RadioGroup;
17  import android.widget.TextView;
18
19  /**
20   * Main activity class for the BMI Calculator program.
21   *
22   * @author Dan Cassidy
23   */
24  public class MainActivity extends Activity {
25      private EditText heightEditText;
26      private EditText weightEditText;
27      private TextView heightAbbreviationTextView;
28      private TextView weightAbbreviationTextView;
29      private TextView bmiResultTextView;
30      private BMICalculator calculatorInstance;
31
32      /**
33       * Main method that runs on application start.
34       *
35       * @param savedInstanceState The saved instance state.
36       */
37      @Override protected void onCreate(Bundle savedInstanceState) {
38          super.onCreate(savedInstanceState);
39          setContentView(R.layout.activity_main);
40
41          heightEditText = (EditText) findViewById(R.id.heightEditText);
42          weightEditText = (EditText) findViewById(R.id.weightEditText);
43          heightAbbreviationTextView = (TextView) findViewById(R.id.heightAbbreviationTextView);
44          weightAbbreviationTextView = (TextView) findViewById(R.id.weightAbbreviationTextView);
45          bmiResultTextView = (TextView) findViewById(R.id.bmiResultTextView);
46          RadioGroup unitsRadioGroup = (RadioGroup) findViewById(R.id.unitsRadioGroup);
47
48          heightEditText.addTextChangedListener(heightEditTextWatcher);
49          weightEditText.addTextChangedListener(weightEditTextWatcher);
50          unitsRadioGroup.setOnCheckedChangeListener(unitsRadioGroupListener);
51
52          calculatorInstance = new BMICalculator();
53
54          updateBMI();
55      }
56
57      /**
58       * Listener for the height field to detect when the input has changed.
59       */
60      private TextWatcher heightEditTextWatcher = new TextWatcher() {

```

```

61         @Override public void onTextChanged(CharSequence s, int start, int before, int count) {
62             try {
63                 calculatorInstance.setHeight(Double.parseDouble(s.toString()));
64             }
65             catch (NumberFormatException e) { }
66             updateBMI();
67         }
68
69         @Override public void beforeTextChanged(CharSequence s, int start, int count, int after) { }
70         @Override public void afterTextChanged(Editable s) { }
71     };
72
73     /**
74     * Listener for the weight field to detect when the input has changed.
75     */
76     private TextWatcher weightEditTextWatcher = new TextWatcher() {
77         @Override public void onTextChanged(CharSequence s, int start, int before, int count) {
78             try {
79                 calculatorInstance.setWeight(Double.parseDouble(s.toString()));
80             }
81             catch (NumberFormatException e) { }
82             updateBMI();
83         }
84
85         @Override public void afterTextChanged(Editable s) { }
86         @Override public void beforeTextChanged(CharSequence s, int start, int count, int after) { }
87     };
88
89     /**
90     * Listener for the RadioGroup to detect when the user has switched from metric to English units
91     * and vice versa.
92     */
93     private RadioGroup.OnCheckedChangeListener unitsRadioGroupListener =
94         new RadioGroup.OnCheckedChangeListener() {
95         @Override public void onCheckedChanged(RadioGroup group, int checkedId) {
96             if (checkedId == R.id.metricRadioButton) {
97                 calculatorInstance.setMetric(true);
98                 heightAbbreviationTextView.setText(R.string.meter_abbreviation);
99                 weightAbbreviationTextView.setText(R.string.kilogram_abbreviation);
100             }
101             else {
102                 calculatorInstance.setMetric(false);
103                 heightAbbreviationTextView.setText(R.string.inch_abbreviation);
104                 weightAbbreviationTextView.setText(R.string.pound_abbreviation);
105             }
106             heightEditText.setText(String.format("%.2f", calculatorInstance.getHeight()));
107             weightEditText.setText(String.format("%.2f", calculatorInstance.getWeight()));
108             updateBMI();
109         }
110     };
111
112     /**
113     * Helper method to update the activity's BMI display when needed.
114     */
115     private void updateBMI() {
116         bmiResultTextView.setText(String.format("%.2f", calculatorInstance.getBMI()));
117     }
118 }
119

```

```
1  /*-----
2  * Author:      Dan Cassidy
3  * Date:        2015-07-24
4  * Assignment:  HW6-1
5  * Source File: BMICalculator.java
6  * Language:    Java
7  * Course:      CSCI-C 490, Android Programming, MoWe 08:00
8  -----*/
9  package dancassidy.bmicalculator;
10
11  /**
12   * A class to calculate a person's body mass index (BMI).
13   *
14   * @author Dan Cassidy
15   */
16  public class BMICalculator {
17      private double weight = 0;
18      private double height = 0;
19      private double bmi = 0;
20      boolean metric = true;
21
22      // BEGIN GETTERS AND SETTERS -->
23      public double getBMI() {
24          return this.bmi;
25      }
26
27      public double getHeight() {
28          return this.height;
29      }
30
31      public void setHeight(double height) {
32          if (height >= 0) {
33              this.height = height;
34              calculateBMI();
35          }
36      }
37
38      public double getWeight() {
39          return this.weight;
40      }
41
42      public void setWeight(double weight) {
43          if (weight >= 0) {
44              this.weight = weight;
45              calculateBMI();
46          }
47      }
48
49      public boolean getMetric() {
50          return metric;
51      }
52
53      public void setMetric(boolean metric) {
54          if (this.metric ^ metric) {
55              if (this.metric) {
56                  this.height = UnitConversion.lengthMetersToInches(this.height);
57                  this.weight = UnitConversion.weightKilogramsToPounds(this.weight);
58              } else {
59                  this.height = UnitConversion.lengthInchesToMeters(this.height);
60                  this.weight = UnitConversion.weightPoundsToKilograms(this.weight);
```

```
61         }
62         this.metric = metric;
63         calculateBMI();
64     }
65 }
66 // <-- END GETTERS AND SETTERS
67
68 /**
69  * Helper method to calculate the BMI.
70  */
71 private void calculateBMI() {
72     if (this.height == 0)
73         this.bmi = 0;
74     else if (this.metric)
75         this.bmi = this.weight / (this.height * this.height);
76     else
77         this.bmi = (this.weight * 703) / (this.height * this.height);
78 }
79 }
80
```

```
1  /*-----*/
2  * Author:      Dan Cassidy
3  * Date:        2015-07-24
4  * Assignment:  HW6-1
5  * Source File: UnitConversion.java
6  * Language:    Java
7  * Course:      CSCI-C 490, Android Programming, MoWe 08:00
8  -----*/
9  package dancassidy.bmicalculator;
10
11 /**
12  * Small utility class to convert between different units.
13  *
14  * @author Dan Cassidy
15  */
16 public class UnitConversion {
17     // Conversion factors.
18     private static final double POUND_KILOGRAM_CONVERSION_FACTOR = 0.45359237;
19     private static final double INCH_METER_CONVERSION_FACTOR = 0.0254;
20
21     /**
22      * Converts kilograms to pounds.
23      *
24      * @param weightInKilograms The weight in kilograms.
25      * @return The weight in pounds.
26      */
27     public static double weightKilogramsToPounds(double weightInKilograms) {
28         return weightInKilograms / POUND_KILOGRAM_CONVERSION_FACTOR;
29     }
30
31     /**
32      * Converts pounds to kilograms.
33      *
34      * @param weightInPounds The weight in pounds.
35      * @return The weight in kilograms.
36      */
37     public static double weightPoundsToKilograms(double weightInPounds) {
38         return weightInPounds * POUND_KILOGRAM_CONVERSION_FACTOR;
39     }
40
41     /**
42      * Converts meters to inches.
43      *
44      * @param lengthInMeters The length in meters.
45      * @return The length in inches.
46      */
47     public static double lengthMetersToInches(double lengthInMeters) {
48         return lengthInMeters / INCH_METER_CONVERSION_FACTOR;
49     }
50
51     /**
52      * Converts inches to meters.
53      *
54      * @param lengthInInches The length in inches.
55      * @return The length in meters.
56      */
57     public static double lengthInchesToMeters(double lengthInInches) {
58         return lengthInInches * INCH_METER_CONVERSION_FACTOR;
59     }
60 }
```

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="fill_parent" android:layout_height="fill_parent"
5      android:useDefaultMargins="false"
6      android:columnCount="3"
7      android:paddingLeft="@dimen/activity_horizontal_margin"
8      android:paddingRight="@dimen/activity_horizontal_margin"
9      android:paddingTop="@dimen/activity_vertical_margin"
10     android:paddingBottom="@dimen/activity_vertical_margin">
11
12     <TextView
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
15         android:text="@string/height"
16         android:id="@+id/heightTextView"
17         android:layout_gravity="center_vertical"
18         android:layout_marginRight="10dp" />
19
20     <EditText
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:id="@+id/heightEditText"
24         android:gravity="right"
25         android:inputType="numberDecimal"
26         tools:ignore="RtlHardcoded"
27         android:layout_gravity="fill_horizontal" />
28
29     <TextView
30         android:layout_width="wrap_content"
31         android:layout_height="wrap_content"
32         android:text="@string/meter_abbreviation"
33         android:id="@+id/heightAbbreviationTextView"
34         android:layout_gravity="center_vertical" />
35
36     <TextView
37         android:layout_width="wrap_content"
38         android:layout_height="wrap_content"
39         android:text="@string/weight"
40         android:id="@+id/weightTextView"
41         android:layout_gravity="center_vertical"
42         android:layout_marginRight="10dp" />
43
44     <EditText
45         android:layout_width="wrap_content"
46         android:layout_height="wrap_content"
47         android:id="@+id/weightEditText"
48         android:gravity="right"
49         android:inputType="numberDecimal"
50         tools:ignore="RtlHardcoded"
51         android:layout_gravity="fill_horizontal" />
52
53     <TextView
54         android:layout_width="wrap_content"
55         android:layout_height="wrap_content"
56         android:text="@string/kilogram_abbreviation"
57         android:id="@+id/weightAbbreviationTextView"
58         android:layout_gravity="center_vertical" />
59
60     <TextView
```

```
61         android:layout_width="wrap_content"
62         android:layout_height="wrap_content"
63         android:text="@string/units"
64         android:id="@+id/unitsTextView"
65         android:layout_gravity="center_vertical"
66         android:layout_marginRight="10dp" />
67
68     <RadioGroup
69         android:layout_width="wrap_content"
70         android:layout_height="wrap_content"
71         android:orientation="horizontal"
72         android:id="@+id/unitsRadioGroup">
73
74         <RadioButton
75             android:layout_width="wrap_content"
76             android:layout_height="wrap_content"
77             android:text="@string/unit_metric"
78             android:id="@+id/metricRadioButton"
79             android:checked="true" />
80
81         <RadioButton
82             android:layout_width="wrap_content"
83             android:layout_height="wrap_content"
84             android:text="@string/unit_english"
85             android:id="@+id/englishRadioButton" />
86     </RadioGroup>
87
88     <TextView
89         android:layout_width="wrap_content"
90         android:layout_height="wrap_content"
91         android:text="@string/bmi"
92         android:id="@+id/bmiTextView"
93         android:layout_column="0" />
94
95     <TextView
96         android:layout_width="wrap_content"
97         android:layout_height="wrap_content"
98         android:id="@+id/bmiResultTextView"
99         android:gravity="right"
100        android:layout_gravity="right" />
101
102     <GridLayout
103         android:layout_width="wrap_content"
104         android:layout_height="wrap_content"
105         android:layout_columnSpan="3"
106         android:columnCount="2"
107         android:layout_marginTop="20dp"
108         android:id="@+id/bmiInfoGridLayout">
109
110         <TextView
111             android:layout_width="wrap_content"
112             android:layout_height="wrap_content"
113             android:text="@string/bmi_info_title"
114             android:id="@+id/bmiInfoTitleTextView"
115             android:layout_columnSpan="2" />
116
117         <TextView
118             android:layout_width="wrap_content"
119             android:layout_height="wrap_content"
120             android:text="@string/bmi_info_underweight"
```



```
121         android:id="@+id/bmiInfoUnderweightTextView"
122         android:paddingRight="10dp" />
123
124     <TextView
125         android:layout_width="wrap_content"
126         android:layout_height="wrap_content"
127         android:text="@string/bmi_info_underweight_details"
128         android:id="@+id/bmiInfoUnderweightDetailsTextView" />
129
130     <TextView
131         android:layout_width="wrap_content"
132         android:layout_height="wrap_content"
133         android:text="@string/bmi_info_normal"
134         android:id="@+id/bmiInfoNormalTextView"
135         android:paddingRight="10dp" />
136
137     <TextView
138         android:layout_width="wrap_content"
139         android:layout_height="wrap_content"
140         android:text="@string/bmi_info_normal_details"
141         android:id="@+id/bmiInfoNormalDetailsTextView" />
142
143     <TextView
144         android:layout_width="wrap_content"
145         android:layout_height="wrap_content"
146         android:text="@string/bmi_info_overweight"
147         android:id="@+id/bmiInfoOverweightTextView"
148         android:paddingRight="10dp" />
149
150     <TextView
151         android:layout_width="wrap_content"
152         android:layout_height="wrap_content"
153         android:text="@string/bmi_info_overweight_details"
154         android:id="@+id/bmiInfoOverweightDetailsTextView" />
155
156     <TextView
157         android:layout_width="wrap_content"
158         android:layout_height="wrap_content"
159         android:text="@string/bmi_info_obese"
160         android:id="@+id/bmiInfoObeseTextView"
161         android:paddingRight="10dp" />
162
163     <TextView
164         android:layout_width="wrap_content"
165         android:layout_height="wrap_content"
166         android:text="@string/bmi_info_obese_details"
167         android:id="@+id/bmiInfoObeseDetailsTextView" />
168 </GridLayout>
169
170 </GridLayout>
171
```

```
1  <resources>
2      <string name="app_name">BMI Calculator</string>
3
4      <string name="height">Height</string>
5      <string name="weight">Weight</string>
6      <string name="pound_abbreviation">lb</string>
7      <string name="kilogram_abbreviation">kg</string>
8      <string name="inch_abbreviation">in</string>
9      <string name="meter_abbreviation">m</string>
10     <string name="units">Units</string>
11     <string name="unit_metric">Metric</string>
12     <string name="unit_english">English</string>
13     <string name="bmi">BMI</string>
14     <string name="bmi_info_title">BMI VALUES</string>
15     <string name="bmi_info_underweight">Underweight:</string>
16     <string name="bmi_info_underweight_details">less than 18.5</string>
17     <string name="bmi_info_normal">Normal:</string>
18     <string name="bmi_info_normal_details">between 18.5 and 24.9</string>
19     <string name="bmi_info_overweight">Overweight:</string>
20     <string name="bmi_info_overweight_details">between 25 and 29.9</string>
21     <string name="bmi_info_obese">Obese:</string>
22     <string name="bmi_info_obese_details">30 or greater</string>
23 </resources>
24
```

Just after startup.

3G

11:32

Height

m

Weight

kg

Units

☒ Metric ☐ English

BMI

0.00

BMI VALUES

Underweight:

less than 18.5

Normal:

between 18.5 and 24.9

Overweight:


between 25 and 29.9

Obese:

30 or greater

Demonstrating that 0's do not cause a crash.

3G



11:35

Height

0 m

Weight

0 kg

Units

☒ Metric ☐ English

BMI

0.00

BMI VALUES

Underweight: less than 18.5

Normal: between 18.5 and 24.9

Overweight: between 25 and 29.9

Obese: 30 or greater

Showing that standard input works.

3G

11:36

Height

2 m

Weight

100 kg

Units

☒ Metric ☐ English

BMI

25.00

BMI VALUES

Underweight: less than 18.5

Normal: between 18.5 and 24.9

Overweight: between 25 and 29.9

Obese: 30 or greater

Showing that unit conversion works and is automatic.

3G

11:36

Height

78.74 in

Weight

220.46 lb

Units

☐ Metric ☒ English

BMI

25.00

BMI VALUES

Underweight: less than 18.5

Normal: between 18.5 and 24.9

Overweight: between 25 and 29.9

Obese: 30 or greater

Confirming that unit conversion works in the other direction as well.

3G

11:37

Height

2.00 m

Weight

100.00 kg

Units

☒ Metric ☐ English

BMI

25.00

BMI VALUES

Underweight: less than 18.5

Normal: between 18.5 and 24.9

Overweight: between 25 and 29.9

Obese: 30 or greater

Name: Dan Cassidy

Class: CSCI-C 490, Mobile Application Development

Assignment: Homework 6 Part 2

Date: 2015-07-25


```

1  /*-----
2  * Author:      Dan Cassidy
3  * Date:        2015-07-25
4  * Assignment:  HW6-2
5  * Source File: MainActivity.java
6  * Language:    Java
7  * Course:      CSCI-C 490, Android Programming, MoWe 08:00
8  -----*/
9  package dancassidy.targetheartratecalculator;
10
11  import android.app.Activity;
12  import android.os.Bundle;
13  import android.text.Editable;
14  import android.text.TextWatcher;
15  import android.widget.EditText;
16  import android.widget.TextView;
17
18  /**
19   * Main activity class for Target Heart Rate Calculator program.
20   *
21   * @author Dan Cassidy
22   */
23  public class MainActivity extends Activity {
24      private TextView targetHeartRateRangeTextView;
25
26      /**
27       * Main method that runs on application start.
28       *
29       * @param savedInstanceState The saved instance state.
30       */
31      @Override protected void onCreate(Bundle savedInstanceState) {
32          super.onCreate(savedInstanceState);
33          setContentView(R.layout.activity_main);
34
35          EditText ageEditText = (EditText) findViewById(R.id.ageEditText);
36          ageEditText.addTextChangedListener(ageEditTextWatcher);
37
38          targetHeartRateRangeTextView =
39              (TextView) findViewById(R.id.targetHeartRateRangeResultTextView);
40      }
41
42      /**
43       * Listener for the age field to detect and handle changes.
44       */
45      private TextWatcher ageEditTextWatcher = new TextWatcher() {
46          @Override public void onTextChanged(CharSequence s, int start, int before, int count) {
47              final int MAX_HEART_RATE = 220;
48              final float LOWER_THR_MULT = 0.5f;
49              final float UPPER_THR_MULT = 0.85f;
50
51              try {
52                  int age = Integer.parseInt(s.toString());
53                  if (age > 220)
54                      age = 220;
55
56                  int lowerTHR = Math.round((MAX_HEART_RATE - age) * LOWER_THR_MULT);
57                  int upperTHR = Math.round((MAX_HEART_RATE - age) * UPPER_THR_MULT);
58                  targetHeartRateRangeTextView.setText(Integer.toString(lowerTHR) + " to " +
59                      Integer.toString(upperTHR));
60              }

```


```
61         catch (NumberFormatException e) {
62             targetHeartRateRangeTextView.setText("");
63         }
64     }
65
66     @Override public void beforeTextChanged(CharSequence s, int start, int count, int after) { }
67     @Override public void afterTextChanged(Editable s) { }
68 };
69 }
70
```

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="fill_parent"
4      android:layout_height="fill_parent"
5      android:orientation="vertical"
6      android:paddingLeft="@dimen/activity_horizontal_margin"
7      android:paddingRight="@dimen/activity_horizontal_margin"
8      android:paddingTop="@dimen/activity_vertical_margin"
9      android:paddingBottom="@dimen/activity_vertical_margin">
10
11      <TextView
12          android:layout_width="wrap_content"
13          android:layout_height="wrap_content"
14          android:textAppearance="?android:attr/textAppearanceLarge"
15          android:text="@string/app_name"
16          android:id="@+id/titleTextView"
17          android:layout_gravity="center" />
18
19      <LinearLayout
20          android:orientation="horizontal"
21          android:layout_width="match_parent"
22          android:layout_height="wrap_content">
23
24          <TextView
25              android:layout_width="wrap_content"
26              android:layout_height="wrap_content"
27              android:text="@string/age"
28              android:id="@+id/ageTextView"
29              android:layout_weight="1" />
30
31          <EditText
32              android:layout_width="wrap_content"
33              android:layout_height="wrap_content"
34              android:id="@+id/ageEditText"
35              android:inputType="number"
36              android:gravity="right"
37              android:maxLength="3"
38              android:layout_weight="1" />
39      </LinearLayout>
40
41      <LinearLayout
42          android:orientation="horizontal"
43          android:layout_width="match_parent"
44          android:layout_height="wrap_content">
45
46          <TextView
47              android:layout_width="wrap_content"
48              android:layout_height="wrap_content"
49              android:text="@string/target_heart_rate"
50              android:id="@+id/targetHeartRateRangeTextView"
51              android:layout_weight="1" />
52
53          <TextView
54              android:layout_width="wrap_content"
55              android:layout_height="wrap_content"
56              android:id="@+id/targetHeartRateRangeResultTextView"
57              android:gravity="right"
58              android:layout_weight="1" />
59
60      </LinearLayout>
```

```
61  
62     </LinearLayout>
```

```
1 <resources>
2   <string name="app_name">Target Heart Rate Calculator</string>
3
4   <string name="target_heart_rate">Target Heart Rate:</string>
5   <string name="age">Age:</string>
6 </resources>
7
```

Basic screen.

3G 2:36

Target Heart Rate Calculator

Age:

Target Heart Rate:

Limit case of 0.

3G

2:37

Target Heart Rate Calculator

Age:

Target Heart Rate: 110 to 187

Limit case of 220.

3G

2:37

Target Heart Rate Calculator

Age:

220

Target Heart Rate:

0 to 0

Shows that outside the range, target heart rate stops at 0.

3G

2:38

Target Heart Rate Calculator


Age:

500

Target Heart Rate:

0 to 0

Normal input.

3G 2:38

Target Heart Rate Calculator

Age:

32

Target Heart Rate:

94 to 160

Name: Dan Cassidy

Class: CSCI-C 490, Mobile Application Development

Assignment: Homework 6 Part 3

Date: 2015-07-25

```

1  /-----
2  * Author:      Dan Cassidy
3  * Date:        2015-07-25
4  * Assignment:  HW6-3
5  * Source File: MainActivity.java
6  * Language:    Java
7  * Course:      CSCI-C 490, Android Programming, MoWe 08:00
8  -----*/
9  package dancassidy.craps;
10
11  import android.app.Activity;
12  import android.os.Bundle;
13  import android.view.View;
14  import android.widget.Button;
15  import android.widget.ImageView;
16  import android.widget.TextView;
17
18  /**
19   * Main activity class for the Craps game.
20   *
21   * @author Dan Cassidy
22   */
23  public class MainActivity extends Activity {
24      private static int[] dieResourceIDs = new int[] {R.drawable.die1, R.drawable.die2,
25          R.drawable.die3, R.drawable.die4, R.drawable.die5, R.drawable.die6};
26
27      private ImageView pointDie1;
28      private ImageView pointDie2;
29      private ImageView auxDie1;
30      private ImageView auxDie2;
31      private TextView pointTextView;
32      private TextView statusTextView;
33      private Button playButton;
34      private Button rollButton;
35
36      private Craps gameInstance;
37
38      /**
39       * Main method that runs on application start.
40       *
41       * @param savedInstanceState The saved instance state.
42       */
43      @Override protected void onCreate(Bundle savedInstanceState) {
44          super.onCreate(savedInstanceState);
45          setContentView(R.layout.activity_main);
46
47          pointDie1 = (ImageView) findViewById(R.id.pointDie1ImageView);
48          pointDie2 = (ImageView) findViewById(R.id.pointDie2ImageView);
49          auxDie1 = (ImageView) findViewById(R.id.auxDie1ImageView);
50          auxDie2 = (ImageView) findViewById(R.id.auxDie2ImageView);
51          pointTextView = (TextView) findViewById(R.id.pointTextView);
52          statusTextView = (TextView) findViewById(R.id.statusTextView);
53          playButton = (Button) findViewById(R.id.playButton);
54          rollButton = (Button) findViewById(R.id.rollButton);
55
56          playButton.setOnClickListener(buttonClickListener);
57          rollButton.setOnClickListener(buttonClickListener);
58
59          gameInstance = new Craps();
60      }

```

```
61
62     /**
63      * Listener for the Play and Roll buttons to detect clicks.
64      */
65     private View.OnClickListener buttonClickListener = new View.OnClickListener() {
66         @Override public void onClick(View v) {
67             if (v == playButton) {
68                 gameInstance.reset();
69                 viewReset();
70             }
71
72             gameInstance.roll();
73
74             if (v == playButton) {
75                 pointDie1.setImageResource(dieResourceIDs[gameInstance.getPointRoll1() - 1]);
76                 pointDie1.setVisibility(View.VISIBLE);
77                 pointDie2.setImageResource(dieResourceIDs[gameInstance.getPointRoll2() - 1]);
78                 pointDie2.setVisibility(View.VISIBLE);
79             }
80             else {
81                 auxDie1.setImageResource(dieResourceIDs[gameInstance.getAuxRoll1() - 1]);
82                 auxDie1.setVisibility(View.VISIBLE);
83                 auxDie2.setImageResource(dieResourceIDs[gameInstance.getAuxRoll2() - 1]);
84                 auxDie2.setVisibility(View.VISIBLE);
85             }
86
87             switch (gameInstance.getGameStatus()) {
88                 case WIN:
89                 case LOSE:
90                     playButton.setEnabled(true);
91                     rollButton.setEnabled(false);
92                     statusTextView.setText(gameInstance.getGameStatus() == Craps.Status.WIN ?
93                         R.string.player_win : R.string.player_lose);
94                     break;
95
96                 case ONGOING:
97                     playButton.setEnabled(false);
98                     rollButton.setEnabled(true);
99                     pointTextView.setText(getString(R.string.point) + ": " +
100                         Integer.toString(gameInstance.getPoint()));
101                     pointTextView.setVisibility(View.VISIBLE);
102                     statusTextView.setText(R.string.player_roll);
103                     break;
104             }
105
106             statusTextView.setVisibility(View.VISIBLE);
107         }
108     };
109
110     /**
111      * Reset the visibility of components to default.
112      */
113     private void viewReset() {
114         pointDie1.setVisibility(View.INVISIBLE);
115         pointDie2.setVisibility(View.INVISIBLE);
116         auxDie1.setVisibility(View.INVISIBLE);
117         auxDie2.setVisibility(View.INVISIBLE);
118         pointTextView.setVisibility(View.INVISIBLE);
119         statusTextView.setVisibility(View.INVISIBLE);
120     }
```

```
121     }  
122
```

```
1  /*-----
2  * Author:      Dan Cassidy
3  * Date:        2015-07-25
4  * Assignment:  HW6-3
5  * Source File: Craps.java
6  * Language:    Java
7  * Course:      CSCI-C 490, Android Programming, MoWe 08:00
8  -----*/
9  package dancassidy.craps;
10
11  import java.util.Random;
12
13  /**
14   * Simple craps game implementation.
15   *
16   * @author Dan Cassidy
17   */
18  public class Craps {
19      public static enum Status {
20          ONGOING,
21          WIN,
22          LOSE
23      }
24
25      private int pointRoll1 = 0;
26      private int pointRoll2 = 0;
27      private int auxRoll1 = 0;
28      private int auxRoll2 = 0;
29      private Status gameStatus = Status.ONGOING;
30
31      private static Random generator = new Random();
32
33      // BEGIN GETTERS -->
34      public int getPointRoll1() {
35          return pointRoll1;
36      }
37
38      public int getPointRoll2() {
39          return pointRoll2;
40      }
41
42      public int getPoint() {
43          return pointRoll1 + pointRoll2;
44      }
45
46      public int getAuxRoll1() {
47          return auxRoll1;
48      }
49
50      public int getAuxRoll2() {
51          return auxRoll2;
52      }
53
54      public Status getGameStatus() {
55          return gameStatus;
56      }
57      // <-- END GETTERS
58
59      /**
60       * Roll the appropriate set of die and handle the results.
```

```
61     */
62     public void roll() {
63         if (gameStatus != Status.ONGOING)
64             return;
65
66         // Assign rolls to their respective variables, then check the results and set the game
67         // status accordingly.
68         if (pointRoll1 == 0) {
69             pointRoll1 = generator.nextInt(6) + 1;
70             pointRoll2 = generator.nextInt(6) + 1;
71
72             switch (pointRoll1 + pointRoll2) {
73                 case 7:
74                 case 11:
75                     gameStatus = Status.WIN;
76                     break;
77
78                 case 2:
79                 case 3:
80                 case 12:
81                     gameStatus = Status.LOSE;
82                     break;
83             }
84         }
85         else {
86             auxRoll1 = generator.nextInt(6) + 1;
87             auxRoll2 = generator.nextInt(6) + 1;
88
89             switch (auxRoll1 + auxRoll2) {
90                 case 7:
91                     gameStatus = Status.LOSE;
92                     break;
93
94                 default:
95                     if (pointRoll1 + pointRoll2 == auxRoll1 + auxRoll2)
96                         gameStatus = Status.WIN;
97                     break;
98             }
99         }
100     }
101
102     /**
103     * Reset the game to its default state.
104     */
105     public void reset() {
106         pointRoll1 = 0;
107         pointRoll2 = 0;
108         auxRoll1 = 0;
109         auxRoll2 = 0;
110         gameStatus = Status.ONGOING;
111     }
112 }
113
```



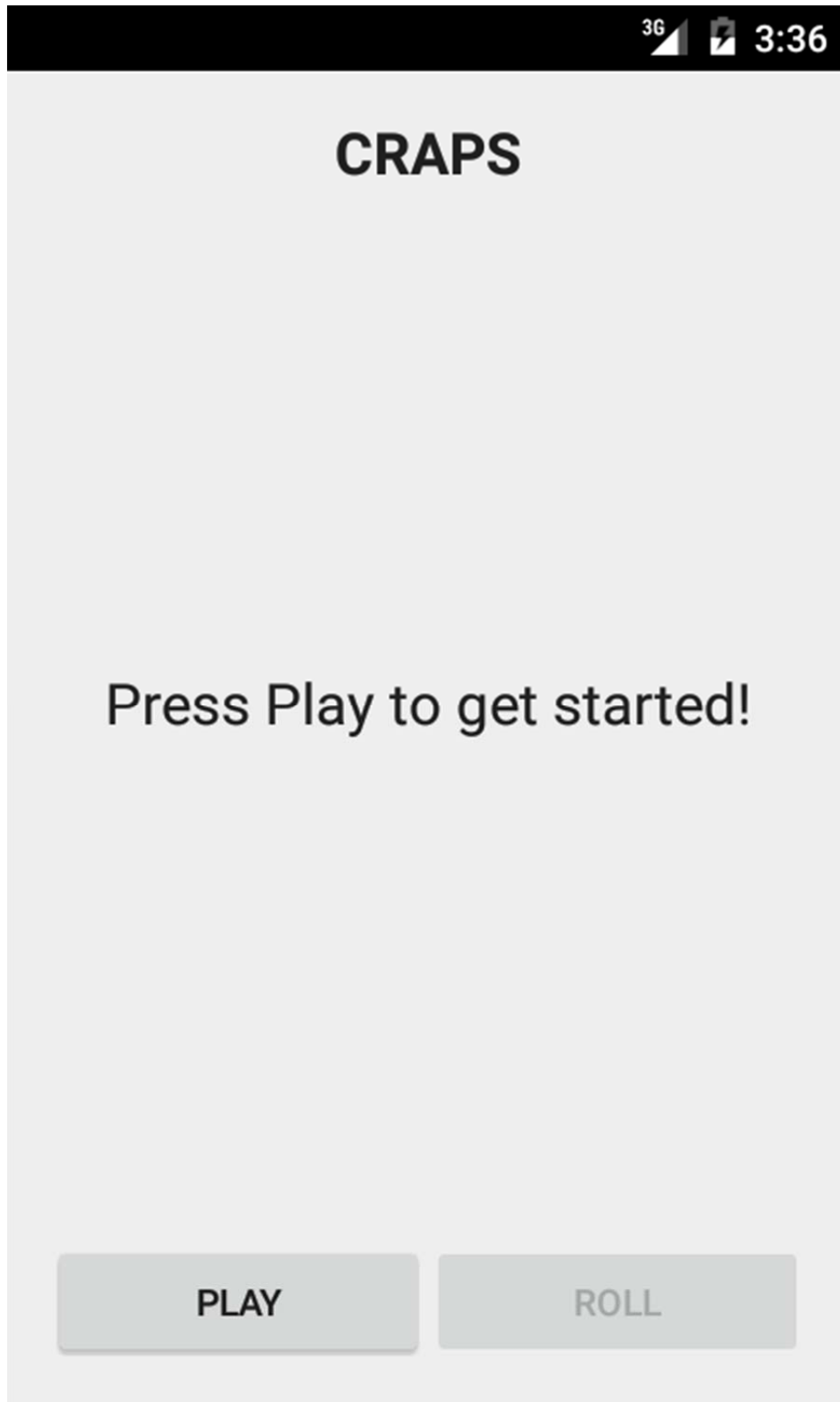
```
1  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
3      android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
4      android:paddingRight="@dimen/activity_horizontal_margin"
5      android:paddingTop="@dimen/activity_vertical_margin"
6      android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">
7
8      <LinearLayout
9          android:orientation="horizontal"
10         android:layout_width="fill_parent"
11         android:layout_height="wrap_content"
12         android:layout_alignParentBottom="true"
13         android:layout_centerHorizontal="true"
14         android:id="@+id/linearLayout">
15
16         <Button
17             android:layout_width="wrap_content"
18             android:layout_height="wrap_content"
19             android:text="@string/play"
20             android:id="@+id/playButton"
21             android:layout_weight="1" />
22
23         <Button
24             android:layout_width="wrap_content"
25             android:layout_height="wrap_content"
26             android:text="@string/roll"
27             android:id="@+id/rollButton"
28             android:layout_weight="1"
29             android:enabled="false" />
30     </LinearLayout>
31
32     <LinearLayout
33         android:orientation="vertical"
34         android:layout_width="match_parent"
35         android:layout_height="wrap_content"
36         android:layout_alignParentTop="true"
37         android:layout_centerHorizontal="true">
38
39         <TextView
40             android:layout_width="wrap_content"
41             android:layout_height="wrap_content"
42             android:textAppearance="?android:attr/textAppearanceLarge"
43             android:text="@string/app_name"
44             android:id="@+id/textView"
45             android:layout_gravity="center"
46             android:textAllCaps="true"
47             android:textStyle="bold" />
48
49         <LinearLayout
50             android:orientation="horizontal"
51             android:layout_width="match_parent"
52             android:layout_height="match_parent"
53             android:gravity="center"
54             android:paddingBottom="20dp"
55             android:paddingTop="20dp">
56
57             <ImageView
58                 android:layout_width="wrap_content"
59                 android:layout_height="wrap_content"
60                 android:id="@+id/pointDie1ImageView"
```

```
61         android:src="@drawable/die1"
62         android:layout_weight="1"
63         android:layout_gravity="center"
64         android:visibility="invisible" />
65
66     <ImageView
67         android:layout_width="wrap_content"
68         android:layout_height="wrap_content"
69         android:id="@+id/pointDie2ImageView"
70         android:src="@drawable/die2"
71         android:layout_weight="1"
72         android:layout_gravity="center"
73         android:visibility="invisible" />
74 </LinearLayout>
75
76 <TextView
77     android:layout_width="wrap_content"
78     android:layout_height="wrap_content"
79     android:textAppearance="?android:attr/textAppearanceLarge"
80     android:id="@+id/pointTextView"
81     android:gravity="center"
82     android:visibility="invisible"
83     android:layout_gravity="center" />
84
85 <LinearLayout
86     android:orientation="horizontal"
87     android:layout_width="match_parent"
88     android:layout_height="match_parent"
89     android:gravity="center"
90     android:paddingTop="20dp"
91     android:paddingBottom="20dp">
92
93     <ImageView
94         android:layout_width="wrap_content"
95         android:layout_height="wrap_content"
96         android:id="@+id/auxDie1ImageView"
97         android:src="@drawable/die3"
98         android:layout_weight="1"
99         android:layout_gravity="center"
100        android:visibility="invisible" />
101
102        <ImageView
103            android:layout_width="wrap_content"
104            android:layout_height="wrap_content"
105            android:id="@+id/auxDie2ImageView"
106            android:src="@drawable/die4"
107            android:layout_weight="1"
108            android:layout_gravity="center"
109            android:visibility="invisible" />
110    </LinearLayout>
111
112    <TextView
113        android:layout_width="wrap_content"
114        android:layout_height="wrap_content"
115        android:textAppearance="?android:attr/textAppearanceLarge"
116        android:id="@+id/statusTextView"
117        android:layout_gravity="center"
118        android:gravity="center"
119        android:text="@string/get_started" />
120 </LinearLayout>
```

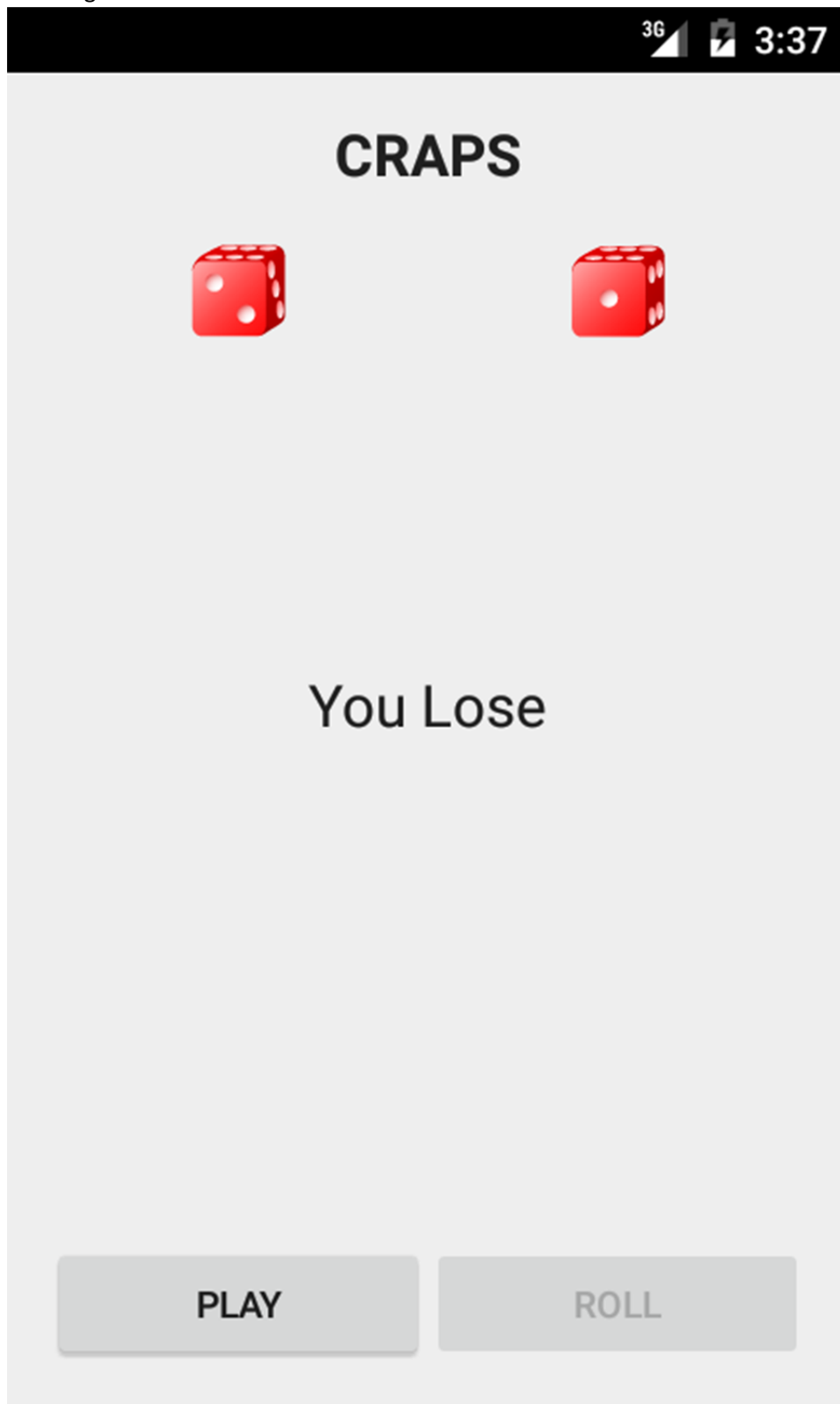
```
121
122     </RelativeLayout>
123
```

```
1  <resources>
2      <string name="app_name">Craps</string>
3
4      <string name="roll">Roll</string>
5      <string name="play">Play</string>
6      <string name="point">Point</string>
7      <string name="player_win">You Win</string>
8      <string name="player_lose">You Lose</string>
9      <string name="player_roll">Roll Again</string>
10     <string name="get_started">Press Play to get started!</string>
11 </resources>
12
```

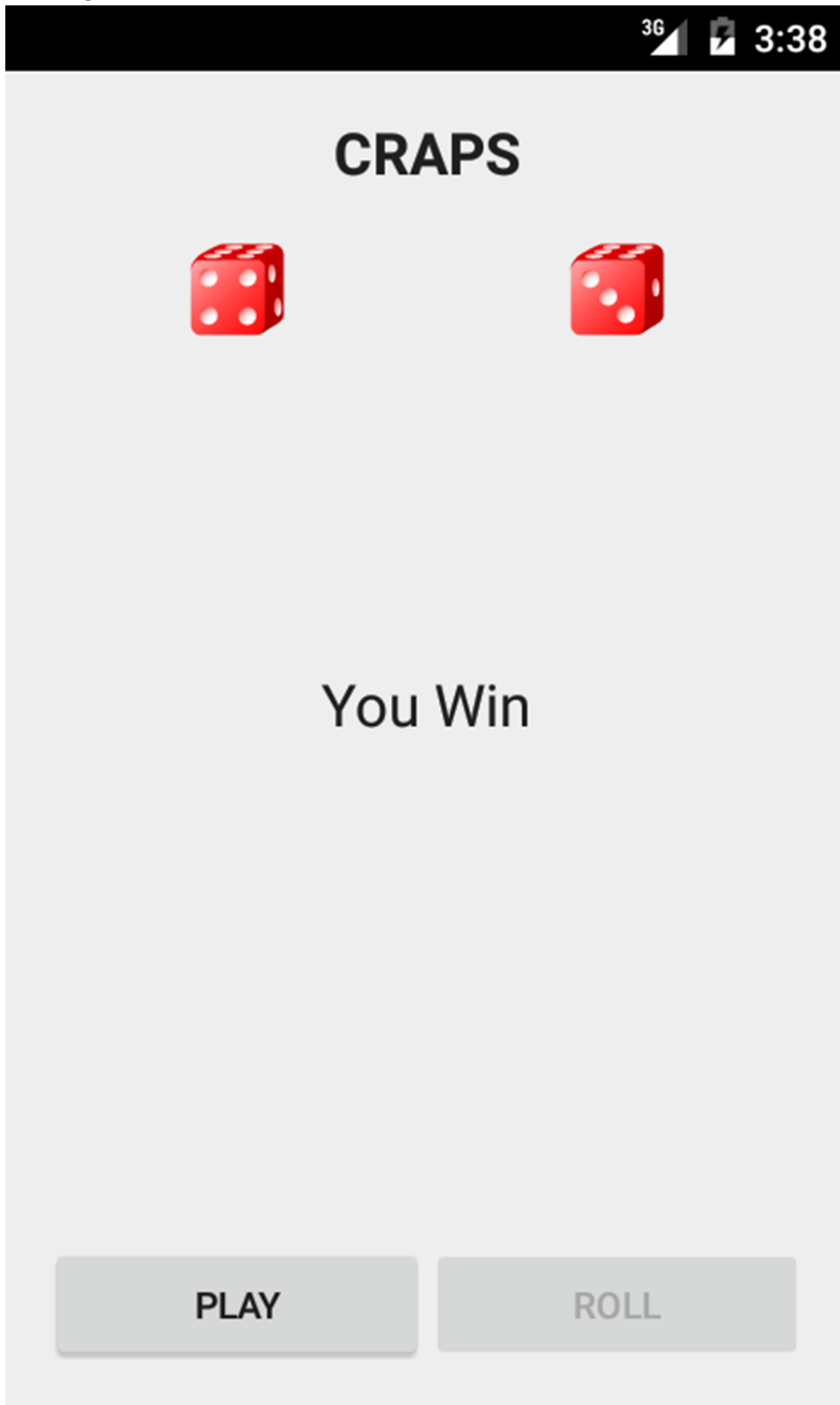
Default screen.



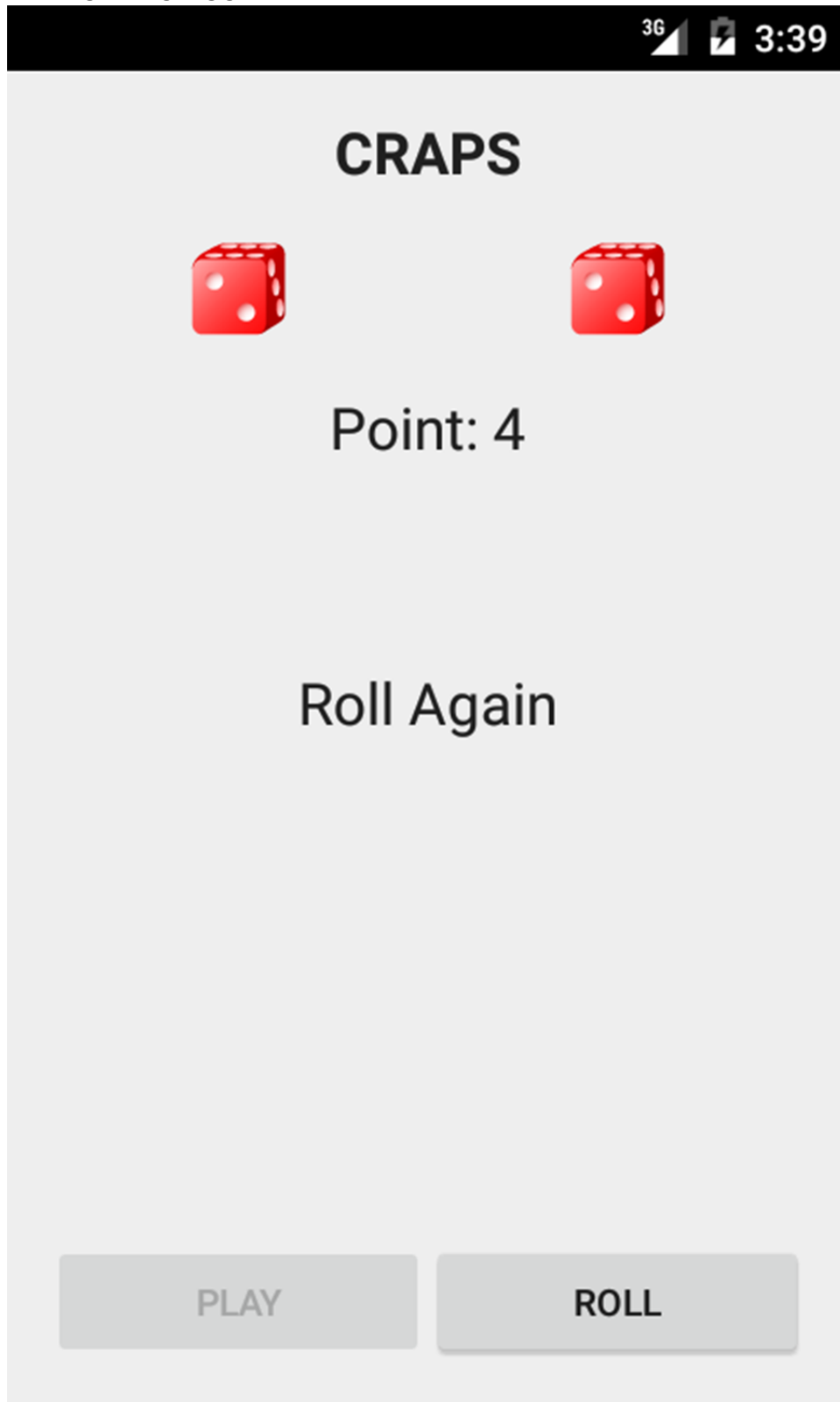
Showing an initial loss.



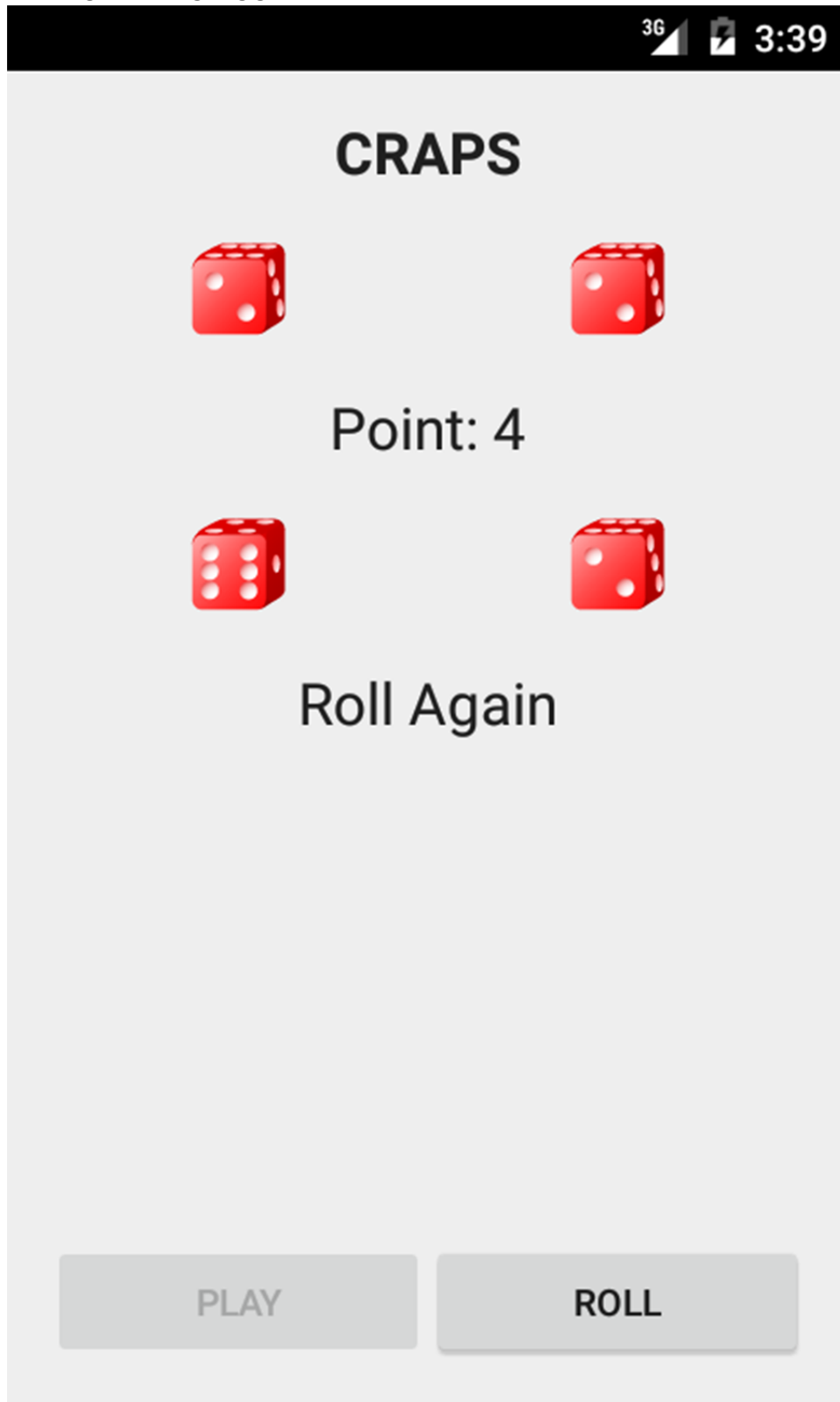
Showing an initial win.



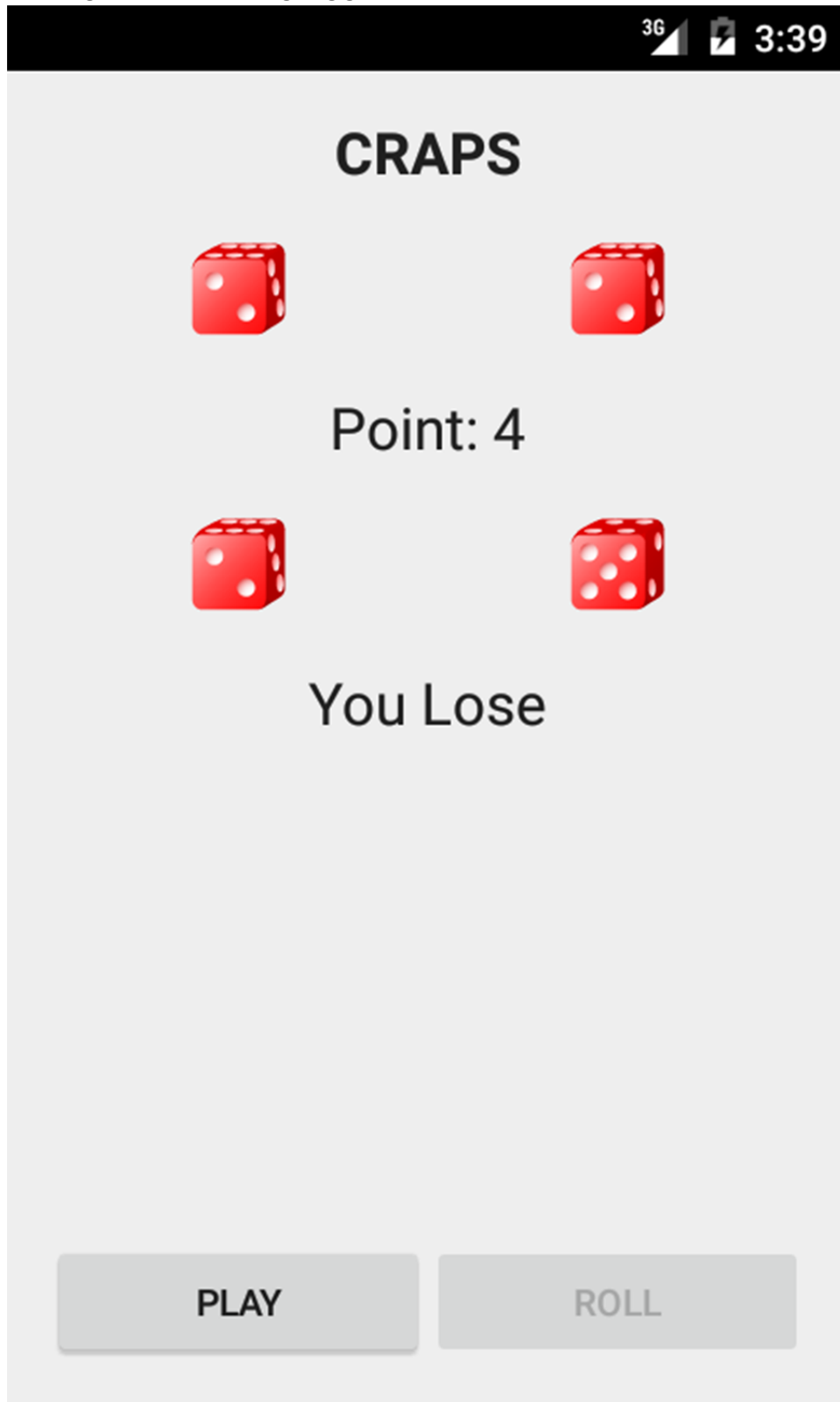
Showing an ongoing game.



Showing a still-ongoing game.



Showing a loss from an ongoing game.



Showing a win from an ongoing game.

