



## الفصل الثاني:

التعامل مع عناصر التحكم الأساسية والقوائم والأحداث

الصفحة	العنوان
3	1. تطبيق 1: آلة حاسبة بسيطة
4	تطبيق 1: بناء آلة حاسبة
5	مراحل العمل
20	2. تطبيق 2: لعبة تعليمية
22	مراحل العمل

## الكلمات المفتاحية:

النسق الخطي LinearLayout، عنصر تحرير النص EditText، عنصر عرض النص TextView، أزرار الأمر Buttons، الأحداث Events، القوائم Menus.

## ملخص:

نعرض في هذا الفصل استخدام عناصر التحكم الأساسية من خلال مثالين تطبيقيين نستخدم فيهما هذه العناصر الأساسية. نُبين التعامل مع عناصر التحكم في الكود البرمجي، الربط مع الأحداث، التعامل مع القوائم.

## أهداف تعليمية:

يتعرف الطالب في هذا الفصل على:

- عناصر التحكم الأساسية.
- التعامل مع الأحداث.
- التعامل مع القوائم.

## المخطط:

التعامل مع عناصر التحكم الأساسية والقوائم والأحداث

- 2 وحدة (Learning Objects)

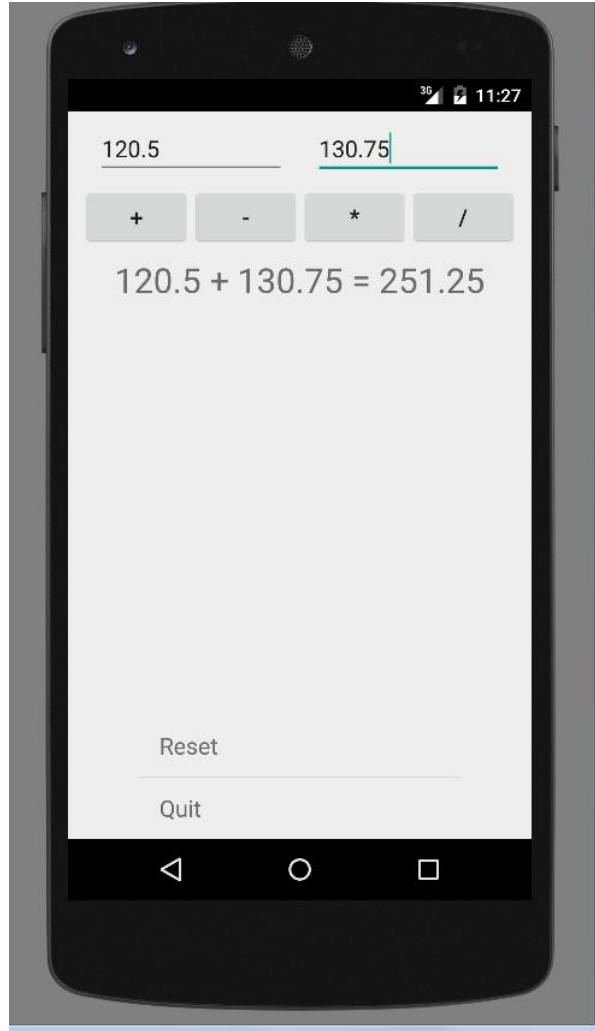
## 1. تطبيق 1: آلة حاسبة بسيطة

الأهداف التعليمية:

- التعامل مع عناصر التحكم الأساسية والقوائم والأحداث

## تطبيق 1: بناء آلة حاسبة

- سنقوم ببناء تطبيق يسمح للمستخدم بإدخال عددين ومن ثم تنفيذ العمليات الحسابية عليهما وإظهار النتيجة. كما يُبين مثلاً التنفيذ التالي:



- كما يوجد للتطبيق قائمة تحوي الخيارين مسح (Reset) لمسح محتوى العناصر وخروج (Quit) لإغلاق التطبيق. (لإظهار القائمة في الجهاز الافتراضي AVD، اضغط F2).

## مراحل العمل

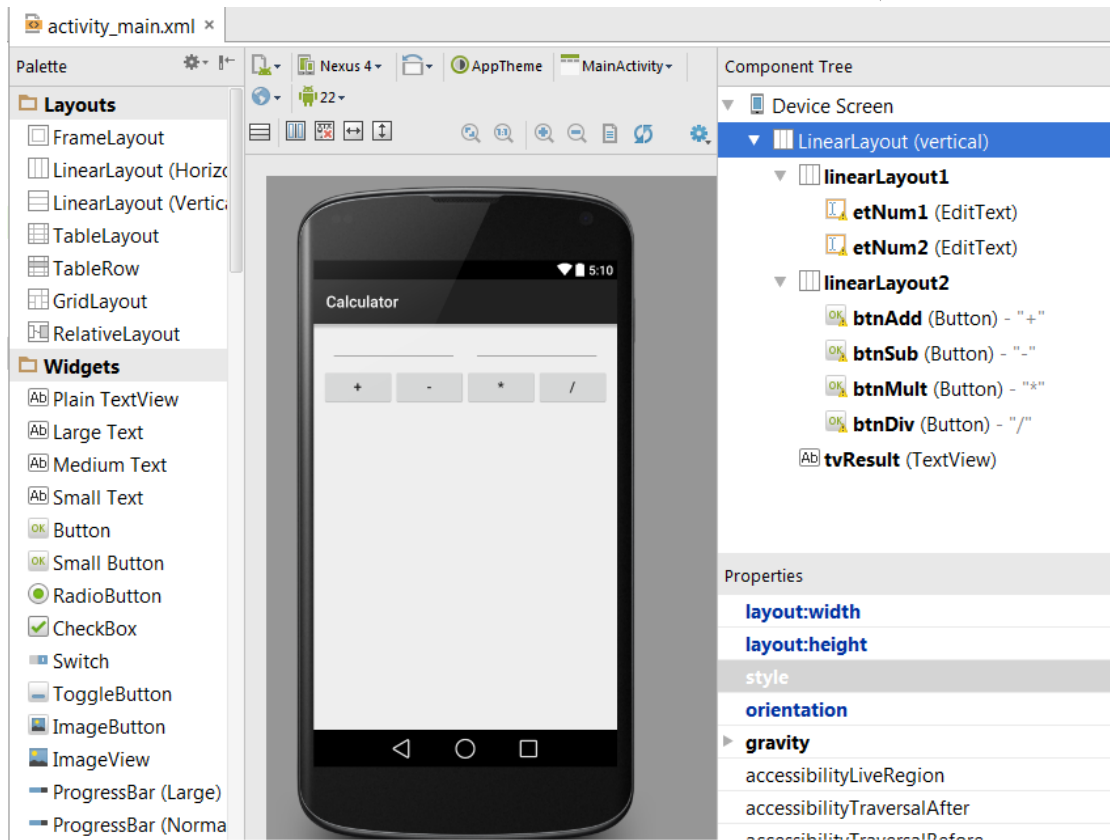
## تصميم النشاط

1. قم بوضع نسق خطي LinearLayout مع ضبط الاتجاه عمودي (orientation="vertical"). يحوي

هذا النسق داخله:

- نسق خطي LinearLayout مع الاتجاه أفقي: يحوي هذا النسق عنصرين من النمط تحرير نص EditText من النوع رقمي عشري Number(Decimal) لإدخال العددين مع ضبط المُعرّفات: etNum1, etNum2.
- نسق خطي LinearLayout مع الاتجاه أفقي: يحوي أربعة أزرار Button مع ضبط المُعرّفات: btnAdd, btnSub, btnMult, btnDiv.
- عنصر عرض نص TextView لإظهار النتيجة مع ضبط المُعرّف tvResult.

وبحيث يُظهر التصميم:



2. قم بربط جميع الأزرار الأربعة مع نفس الإجراءية Calc والتي ستقوم بالحساب (عن طريق الخاصية .onClick).

يكون ملف تصميم النشاط في النهاية :activity\_main.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:orientation="vertical"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent"

    android:paddingBottom="@dimen/activity_vertical_margin"

    tools:context=".MainActivity">

    <LinearLayout

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:id="@+id/linearLayout1"

        android:layout_marginLeft="10pt"

        android:layout_marginRight="10pt"

        android:layout_marginTop="3pt">

        <EditText
```

```
        android:layout_weight="1"

        android:layout_height="wrap_content"

        android:layout_marginRight="5pt"

        android:id="@+id/etNum1"

        android:layout_width="match_parent"

        android:inputType="numberDecimal">

</EditText>

<EditText

        android:layout_height="wrap_content"

        android:layout_weight="1"

        android:layout_marginLeft="5pt"

        android:id="@+id/etNum2"

        android:layout_width="match_parent"

        android:inputType="numberDecimal">

</EditText>

</LinearLayout>

<LinearLayout
```



```
        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:id="@+id/linearLayout2"

        android:layout_marginTop="3pt"

        android:layout_marginLeft="5pt"

        android:layout_marginRight="5pt">

<Button

        android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:layout_weight="1"

        android:text="+"

        android:textSize="8pt"

        android:id="@+id/btnAdd"

        android:onClick="Calc">

</Button>

<Button

        android:layout_height="wrap_content"
```

```
        android:layout_width="match_parent"

        android:layout_weight="1"

        android:text="-"

        android:textSize="8pt"

        android:id="@+id/btnSub"

        android:onClick="Calc">

</Button>

<Button

        android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:layout_weight="1"

        android:text="*"

        android:textSize="8pt"

        android:id="@+id/btnMult"

        android:onClick="Calc">

</Button>

<Button
```

```
        android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:layout_weight="1"

        android:text="/"

        android:textSize="8pt"

        android:id="@+id/btnDiv"

        android:onClick="Calc">

</Button>

</LinearLayout>

<TextView

        android:layout_height="wrap_content"

        android:layout_width="match_parent"

        android:layout_marginLeft="5pt"

        android:layout_marginRight="5pt"

        android:textSize="12pt"

        android:layout_marginTop="3pt"

        android:id="@+id/tvResult"
```

```

        android:gravity="center_horizontal">

    </TextView>

</LinearLayout>

```

### كتابة كود النشاط

3. قم في ملف النشاط بتعريف المتغيرات الموافقة لعناصر التحكم في الصف MainActivity (لأننا سنستخدم هذه المتغيرات في أكثر من إجرائية):

```

EditText etNum1;

EditText etNum2;

Button btnAdd;

Button btnSub;

Button btnMult;

Button btnDiv;

TextView tvResult;

```

4. قم بالوصول إلى عناصر التحكم المختلفة وإسنادها إلى المتغيرات السابقة في الإجرائية onCreate للصف MainActivity:

```

etNum1 = (EditText) findViewById(R.id.etNum1);

etNum2 = (EditText) findViewById(R.id.etNum2);

btnAdd = (Button) findViewById(R.id.btnAdd);

btnSub = (Button) findViewById(R.id.btnSub);

btnMult = (Button) findViewById(R.id.btnMult);

btnDiv = (Button) findViewById(R.id.btnDiv);

tvResult = (TextView) findViewById(R.id.tvResult);

```

### 5. قم بكتابة الإجرائية Calc:

- لاحظ أنه في حال ترك أحد صندوقي التحرير فارغاً، فيتم الخروج من الإجرائية وبدون الحساب.
- يتم اختبار أن صندوق تحرير فارغ باستخدام:

```
TextUtils.isEmpty(etNum1.getText().toString())
```

- بما أن الإجرائية نفسها تُستدعى مع الأزرار الأربعة، نقوم بمعرفة الزر الذي تمّ النقر عليه عن طريق التعليمة:

```
v.getId()
```

وحيث v هو متغير الدخّل للإجرائية Calc من النوع View والذي يدل على الكائن الذي وقع عليه الحدث.

- تكون إذاً إجرائية الحساب المرتبطة مع الأزرار:

```
public void Calc(View v) {

    float num1 = 0;
    float num2 = 0;
    float result = 0;

    // check if the fields are empty
    if (TextUtils.isEmpty(etNum1.getText().toString())
        || TextUtils.isEmpty(etNum2.getText().toString())) {
        return;
    }

    // read EditText and fill variables with numbers
    num1 = Float.parseFloat(etNum1.getText().toString());
    num2 = Float.parseFloat(etNum2.getText().toString());

    // defines the button that has been clicked and performs the corresponding
    operation
    // write operation into oper, we will use it later for output

    switch (v.getId()) {

        case R.id.btnAdd:

            oper = "+";

            result = num1 + num2;
```

```

        break;

    case R.id.btnSub:

        oper = "-";

        result = num1 - num2;

        break;

    case R.id.btnMult:

        oper = "*";

        result = num1 * num2;

        break;

    case R.id.btnDiv:

        oper = "/";

        result = num1 / num2;

        break;

    default:

        break;

}

// form the output line

tvResult.setText(num1 + " " + oper + " " + num2 + " = " + result);

}

```

## 6. قم بإعادة كتابة إجرائية إنشاء قائمة الواجهة :onCreateOptionsMenu

```

public boolean onCreateOptionsMenu(Menu menu) {

    menu.add(0, MENU_RESET_ID, 0, "Reset");

    menu.add(0, MENU_QUIT_ID, 0, "Quit");

    return super.onCreateOptionsMenu(menu);

}

```

- يتم إضافة عنصر للقائمة باستخدام الطريقة add والتي لها أربعة معاملات:

`add(int groupId, int itemId, int order, CharSequence title)`

- تمّ في الصف تعريف الثوابت التالية لعنصري القائمة:

```
final int MENU_RESET_ID = 1;
```

```
final int MENU_QUIT_ID = 2;
```

7. قم بإعادة كتابة إجراءات التفاعل مع عناصر القائمة :onOptionsItemSelected

```
public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

        case MENU_RESET_ID:

            // clear the fields

            etNum1.setText("");

            etNum2.setText("");

            tvResult.setText("");

            break;

        case MENU_QUIT_ID:

            // exit the application

            finish();

            break;

    }

    return super.onOptionsItemSelected(item);

}
```

- تقوم الإجراءات finish() بإنهاء التطبيق.

يكون ملف كود النشاط في النهاية MainActivity.java:

```
package com.example.basel.calculator;

import android.app.Activity;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.text.TextUtils;
```

```
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends Activity {

    EditText etNum1;
    EditText etNum2;
    Button btnAdd;
    Button btnSub;
    Button btnMult;
    Button btnDiv;
    TextView tvResult;
    String oper = "";
    final int MENU_RESET_ID = 1;
    final int MENU_QUIT_ID = 2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```



```
// find the elements

    etNum1 = (EditText) findViewById(R.id.etNum1);

    etNum2 = (EditText) findViewById(R.id.etNum2);

    btnAdd = (Button) findViewById(R.id.btnAdd);

    btnSub = (Button) findViewById(R.id.btnSub);

    btnMult = (Button) findViewById(R.id.btnMult);

    btnDiv = (Button) findViewById(R.id.btnDiv);

    tvResult = (TextView) findViewById(R.id.tvResult);

}

public void Calc(View v) {

    float num1 = 0;

    float num2 = 0;

    float result = 0;

// check if the fields are empty

    if (TextUtils.isEmpty(etNum1.getText().toString())
        || TextUtils.isEmpty(etNum2.getText().toString())) {

        return;

    }

// read EditText and fill variables with numbers

    num1 = Float.parseFloat(etNum1.getText().toString());

    num2 = Float.parseFloat(etNum2.getText().toString());

// defines the button that has been clicked and performs the corresponding operation
```

```
// write operation into oper, we will use it later for output

    switch (v.getId()) {

        case R.id.btnAdd:

            oper = "+";

            result = num1 + num2;

            break;

        case R.id.btnSub:

            oper = "-";

            result = num1 - num2;

            break;

        case R.id.btnMult:

            oper = "*";

            result = num1 * num2;

            break;

        case R.id.btnDiv:

            oper = "/";

            result = num1 / num2;

            break;

        default:

            break;

    }

// form the output line

    tvResult.setText(num1 + " " + oper + " " + num2 + " = " + result);

}

@Override
```

```
// menu creation

public boolean onCreateOptionsMenu(Menu menu) {

    menu.add(0, MENU_RESET_ID, 0, "Reset");

    menu.add(0, MENU_QUIT_ID, 0, "Quit");

    return super.onCreateOptionsMenu(menu);
}

// process menu item clicks

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

        case MENU_RESET_ID:

            // clear the fields

            etNum1.setText("");

            etNum2.setText("");

            tvResult.setText("");

            break;

        case MENU_QUIT_ID:

            // exit the application

            finish();

            break;

    }

    return super.onOptionsItemSelected(item);
}
```

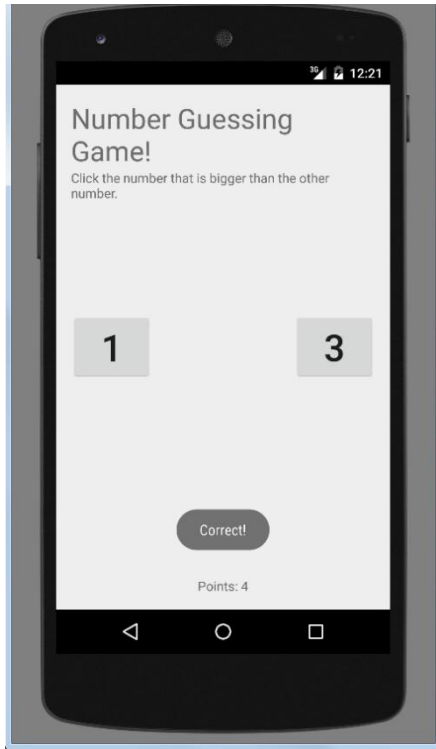
```
}  
}
```

## 2. تطبيق 2: لعبة تعليمية

الأهداف التعليمية:

- التعامل مع عناصر التحكم الأساسية، القوائم، الأحداث

- سنقوم ببناء لعبة بسيطة يُطلب فيها من المستخدم النقر على الرقم الأكبر من رقمين عشوائيين يظهران له على الشاشة.



- في حال نقر المستخدم على الرقم الأكبر تظهر له رسالة منبثقة Correct (صحيح) كما يقوم عدّاد موجود في أسفل الشاشة بجمع عدد مرات الإجابة الصحيحة.
- أما في حال نقر المستخدم على الرقم الأصغر، فتظهر له رسالة موافقة Wrong (خطأ) ويتم إنقاص العداد واحد.
- وفي كلا الحالتين تستمر اللعبة بإظهار أرقام جديدة للمستخدم. وهكذا ...
- كما يُبين مثلاً التنفيذ التالي:

## مراحل العمل

1. قم بتصميم النشاط التالي والذي يحوي على ثلاثة عناصر تحكم من النوع عرض نص TextView وزري أمر Button.

2. قم بضبط خصائص عنصر عرض النص الثالث كما يلي ليظهر أسفل وسط الشاشة:

```
text="Points: 0"
id="@+id/pointsTextView"
layout_alignParentBottom="true"
layout_centerHorizontal="true"
```

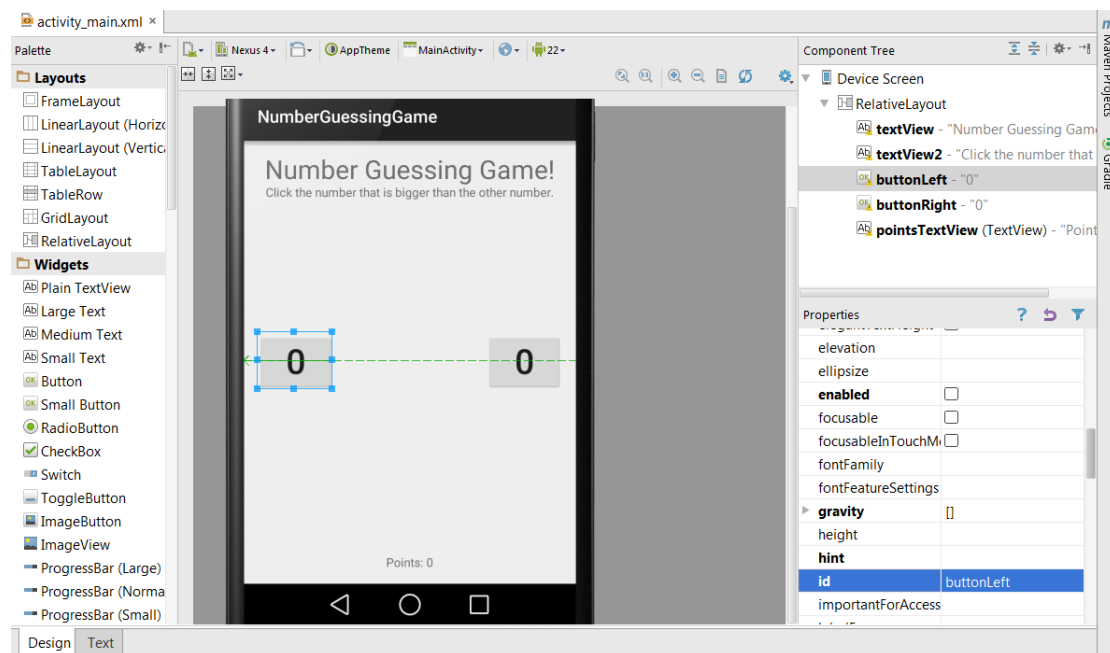
3. قم بضبط خصائص زري الأمر ليظهرها يمين ويسار الشاشة كما يلي:  
الزر اليساري:

```
text="0"
id="@+id/buttonLeft"
layout_centerVertical="true"
layout_alignParentLeft="true"
layout_alignParentStart="true"
textSize="40dp"
onClick="clickButton2"
```

الزر اليميني:

```
text="0"
id="@+id/buttonRight"
layout_centerVertical="true"
layout_alignParentRight="true"
layout_alignParentEnd="true"
textSize="40dp"
onClick="clickButton2"
```

وبحيث يُظهر التصميم:



• يكون ملف تصميم النشاط activity\_main.xml:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"

    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">

    <TextView

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="Number Guessing Game!"
```



```
    android:id="@+id/textView"

    android:layout_alignParentTop="true"

    android:layout_centerHorizontal="true"

    android:textSize="30dp" />
```

#### <TextView

```
    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Click the number that is bigger than the other."

    android:id="@+id/textView2"

    android:layout_below="@+id/textView"

    android:layout_centerHorizontal="true" />
```

#### <Button

```
    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="0"

    android:id="@+id/buttonLeft"

    android:layout_centerVertical="true"

    android:layout_alignParentLeft="true"

    android:layout_alignParentStart="true"

    android:textSize="40dp"

    android:onClick="clickButton1" />
```

<Button

```

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="0"

    android:id="@+id/buttonRight"

    android:layout_centerVertical="true"

    android:layout_alignParentRight="true"

    android:layout_alignParentEnd="true"

    android:textSize="40dp"

    android:onClick="clickButton2" />

```

<TextView

```

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Points: 0"

    android:id="@+id/pointsTextView"

    android:layout_alignParentBottom="true"

    android:layout_centerHorizontal="true"

    android:layout_alignParentEnd="false" />

```

</RelativeLayout>

4. قم في ملف الكود بتعريف المتغيرات التالية في الصف MainActivity:

```

private int num1; //the numbers on the left and right buttons

private int num2;

private int points; // player's point total; initially 0

```

5. قم بكتابة الإجرائية التالية `roll()` والتي تقوم بتوليد رقمين عشوائيين مختلفين بين 0 و 9 ووضع كل منهما على زر أمر:

```
private void roll() {  
  
    // pick two random numbers  
  
    Random r = new Random();  
  
    num1 = r.nextInt(9);  
  
    num2 = r.nextInt(9);  
  
    while (num2 == num1) {  
        num2 = r.nextInt(9);  
    }  
  
    // set the buttons to display the random numbers  
  
    Button left = (Button) findViewById(R.id.buttonLeft);  
    left.setText("" + num1); // "" + int -> converts int to String  
  
    Button right = (Button) findViewById(R.id.buttonRight);  
    right.setText("" + num2);  
  
}
```

6. قم بكتابة الإجرائية `check` والتي تُمرر لها عددين: في حال كان العدد الأول أكبر من الثاني تُظهر رسالة منبثقة بعنوان **Correct!** كما نزيد العداد `points` بمقدار 1. أما في حال كان العدد الأول أصغر من الثاني فنُظهر رسالة منبثقة بعنوان **Wrong!** كما يتم إنقاص العداد `points` بمقدار 1. ثم يتم إظهار قيمة العداد في العنصر `pointsTextView`.

```
private void check(int a, int b) {

    if (a > b) {

        points++;

        Toast.makeText(this, "Correct!",
                        Toast.LENGTH_SHORT).show();

    } else {

        points--;

        Toast.makeText(this, "Wrong!",
                        Toast.LENGTH_SHORT).show();

    }

    TextView pointsView = (TextView)
        findViewById(R.id.pointsTextView);

    pointsView.setText("Points: " + points);

    roll();

}
```

7. قم بإضافة استدعاء الإجرائية `roll()` في الإجرائية `onCreate`:

```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    roll();    // <-- we added this line to set initial button random numbers

}
```

8. قم بكتابة إجرائية الزر الأول والتي تقوم باستدعاء الإجرائية `check` مع تمرير العدد الأول كمعامل أول لها والعدد الثاني كمعامل ثاني لها:

```
public void clickButton1(View view) {
    check(num1, num2);
}
```

9. قم بكتابة إجرائية الزر الثاني والتي تقوم باستدعاء الإجرائية `check` مع تمرير العدد الثاني كمعامل أول لها والعدد الأول كمعامل ثاني لها:

```
public void clickButton2(View view) {
    check(num2, num1);
}
```

10. اربط إجرائية كل زر مع حدث النقر `onClick` للزر.

يكون ملف الكود في النهاية `MainActivity.java`:

```
package com.example.basel.numberguessinggame;

import android.app.*;
import android.support.v7.app.*;
import android.os.*;
import android.view.*;
import android.widget.*;
import java.util.*;

public class MainActivity extends Activity {

    private int num1;    // the numbers on the left and right buttons
    private int num2;
    private int points;  // player's point total; initially 0
}
```

```
/*
 * Called when the player clicks the left number button.
 */

public void clickButton1(View view) {
    check(num1, num2);
}

/*
 * Called when the player clicks the right number button.
 */

public void clickButton2(View view) {
    check(num2, num1);
}

/*
 * Updates the player's score based on whether they guessed correctly.
 * Also shows a 'toast' which is a brief popup message.
 */

private void check(int a, int b) {
    if (a > b) {
        points++;
        Toast.makeText(this, "Correct!",
            Toast.LENGTH_SHORT).show();
    } else {
        points--;
        Toast.makeText(this, "Wrong!",
```

```
        Toast.LENGTH_SHORT).show();

    }

    TextView pointsView = (TextView)
        findViewById(R.id.pointsTextView);

    pointsView.setText("Points: " + points);

    roll();

}

/*
 * Chooses new random integers to appear on the two buttons.
 */

private void roll() {

    // pick two random numbers

    Random r = new Random();

    num1 = r.nextInt(9);

    num2 = r.nextInt(9);

    while (num2 == num1) {

        num2 = r.nextInt(9);

    }

    // set the buttons to display the random numbers

    Button left = (Button) findViewById(R.id.buttonLeft);

    left.setText("" + num1);    // "" + int : converts int to String
}
```

```
        Button right =(Button) findViewById(R.id.buttonRight);  
        right.setText("" + num2);  
    }  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        roll();    // <-- we added this line to set initial button random numbers  
    }  
}
```