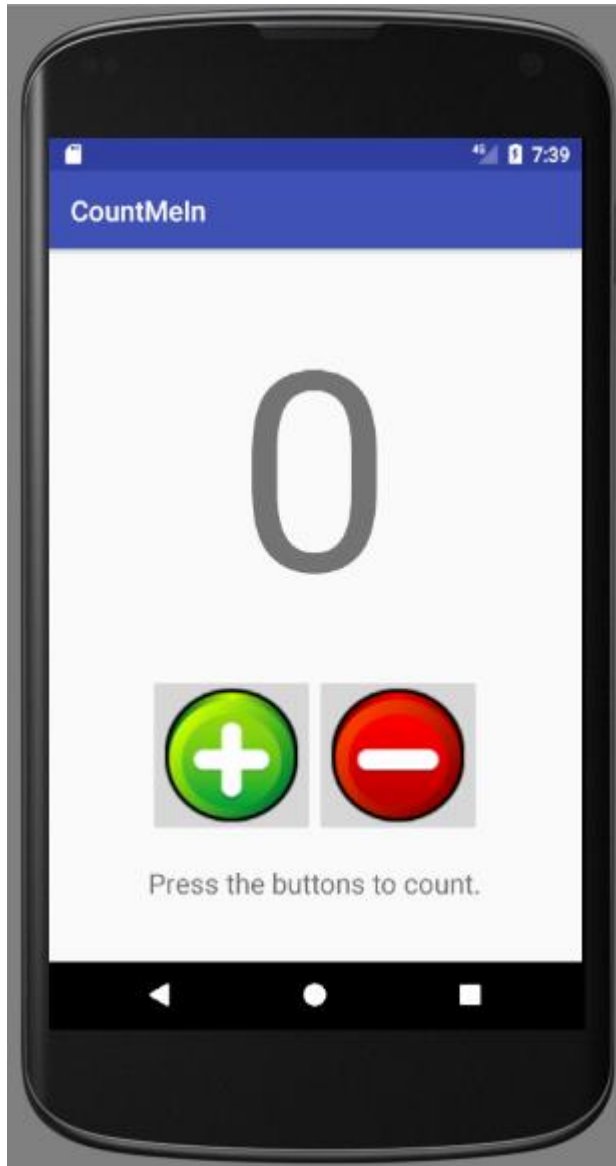
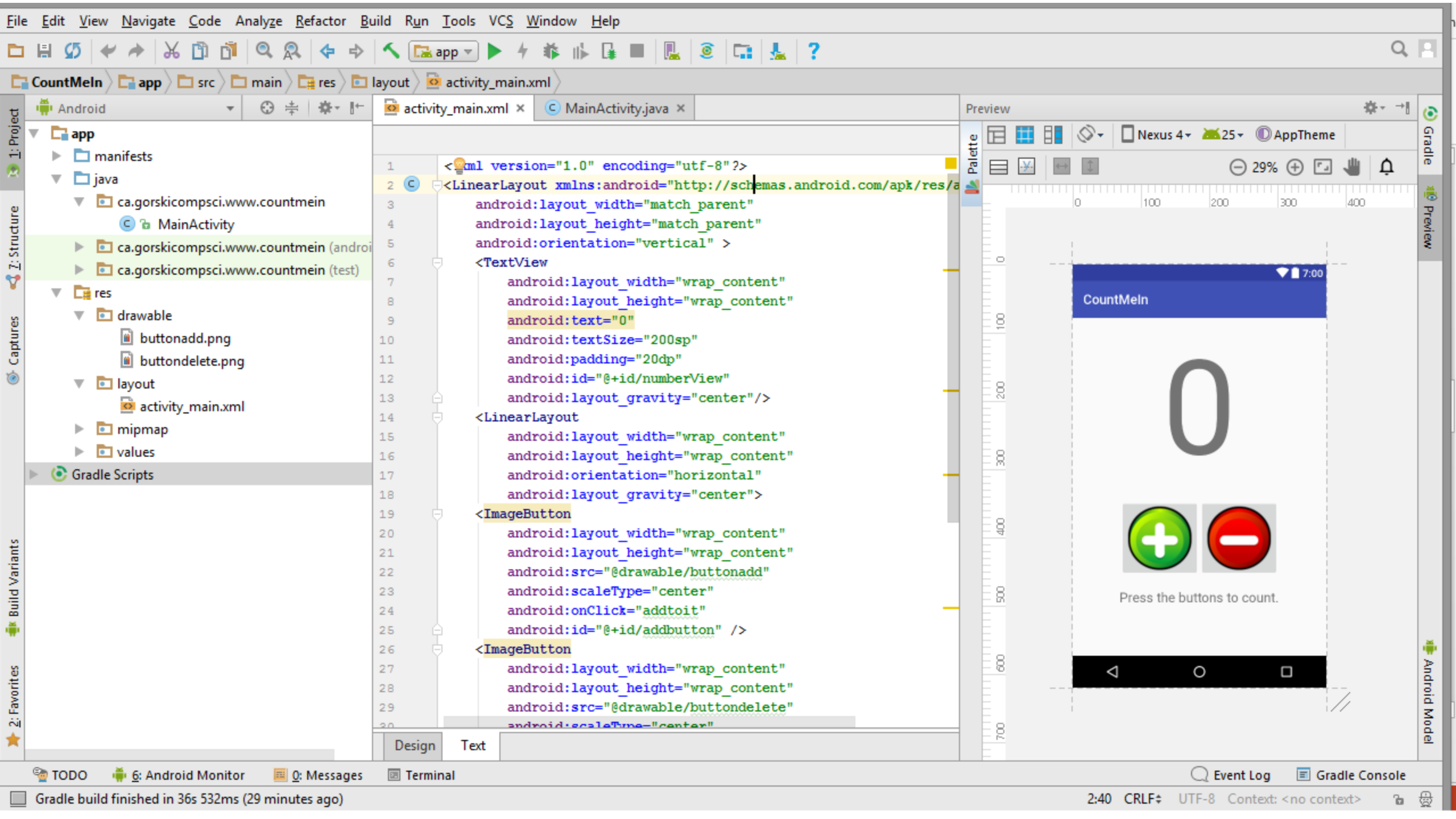
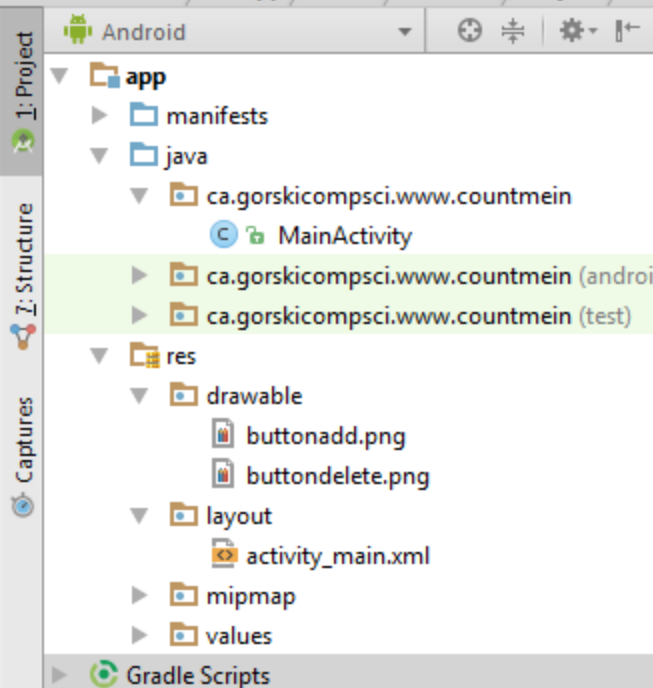


OnClick & Inflation

Buttons that work







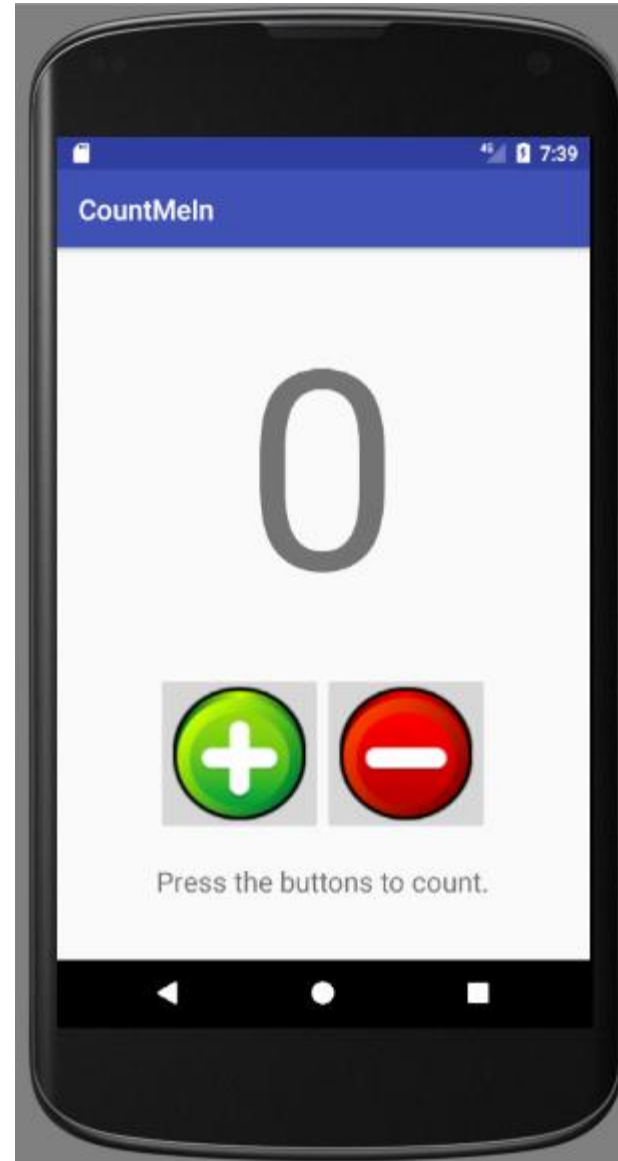
```
1 package ca.gorskicompsci.www.countmein;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.TextView;
7
8 public class MainActivity extends AppCompatActivity {
9     int count = 0;
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14    }
15    public void subtract(View view){
16        TextView txt = (TextView) findViewById(R.id.numberView);
17        count--;
18        txt.setText(""+count);
19    }
20    public void addtoit(View view){
21        TextView txt = (TextView) findViewById(R.id.numberView);
22        count++;
23        txt.setText(""+count);
24    }
25 }
26
```

If it is going to change, it needs an id so the Java code can find it.

`<TextView`

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="0"  
    android:textSize="200sp"  
    android:padding="20dp"  
    android:id="@+id/numberView"  
    android:layout_gravity="center" />
```

View
name



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

Normal variable naming rules apply.

No special
characters

No keywords

No numbers at
the beginning

No spaces

Unlike pictures in android:

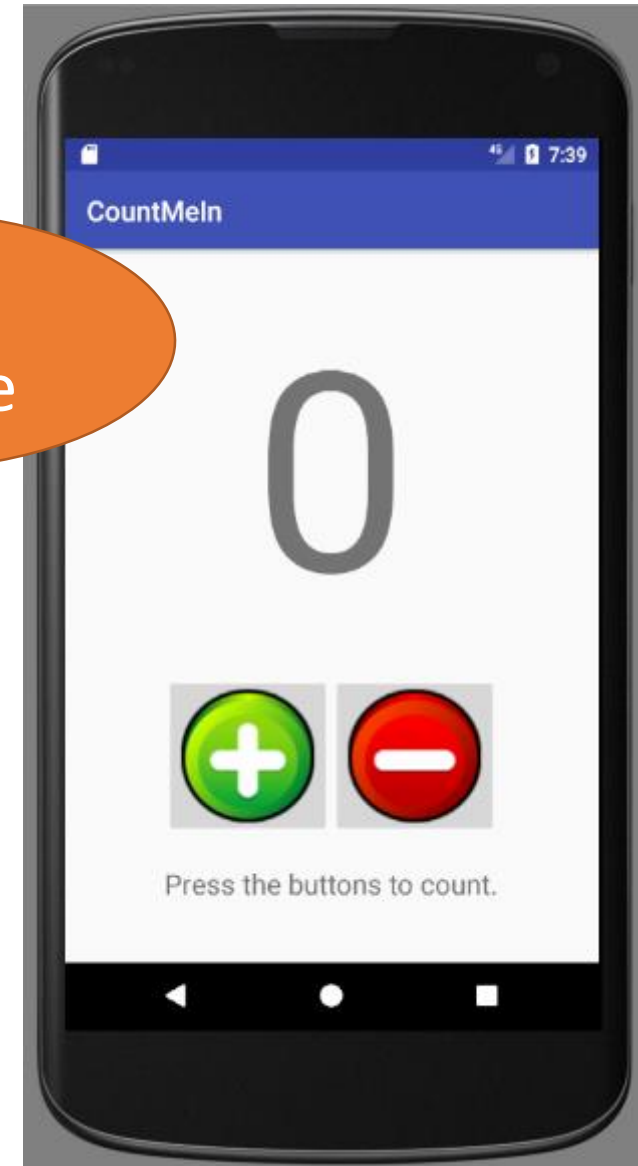
Capitals are
fine

If it is going to be clickable, you need to add an onClick to the Image, Button or ImageButton.

<ImageButton

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/buttonadd"  
    android:scaleType="center"  
    android:onClick="addtoit"  
    android:id="@+id/addbutton" />
```

Make it
clickable



```
android:onClick="addtoit"
```

The onClick is used to make method inside java:

```
public void addToit(View view) {  
    TextView txt=(TextView) findViewById(R.id.numberView) ;  
    count++;  
    txt.setText (" "+count) ;  
}
```



```
android:onClick="addtoit"
```

Normal method naming rules apply.

No special
characters

No keywords

No numbers at
the beginning

No spaces

Unlike pictures in android:

Capitals are
fine

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="0"
    android:textSize="200sp"
    android:padding="20dp"
    android:id="@+id/numberView"
    android:layout_gravity="center"/>

```

View
name

```

<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/buttonadd"
    android:scaleType="center"
    android:onClick="addtoit"
    android:id="@+id/addbutton" />

```

Connect
to
method

```

<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/buttondelete"
    android:scaleType="center"
    android:onClick="subtract"
    android:id="@+id/subtractbutton" />

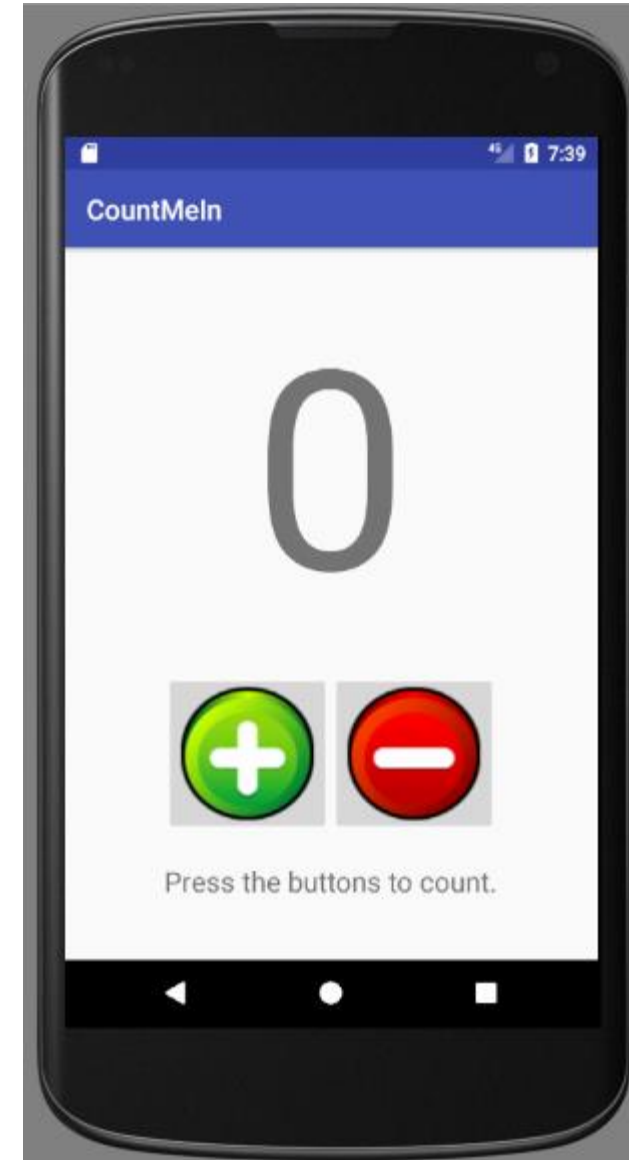
```

Connect
to
method

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Press the buttons to count."
    android:textSize="20sp"
    android:padding="20sp"
    android:layout_gravity="center" />

```



```
package ca.gorskicompsci.www.countmein;
```

```
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.TextView;
```

Still Libraries

```
public class MainActivity extends AppCompatActivity {  
    int count = 0;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
    public void subtract(View view) {  
        TextView txt = (TextView) findViewById(R.id.numberView);  
        count--;  
        txt.setText("" + count);  
    }  
    public void addtoit(View view) {  
        TextView txt = (TextView) findViewById(R.id.numberView);  
        count++;  
        txt.setText("" + count);  
    }  
}
```

Essentially: make
my XML file into
the init method.

The code for my
subtract button.

The code for my
add button.

```
package ca.gorskicompsci.www.countmein;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.TextView;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    int count = 0;
```

Global Variable

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

Method name
matches onclick

```
    public void subtract(View view) {
```

```
        TextView txt = (TextView) findViewById(R.id.numberView);
```

Needed to find
the textview in
the XML

Change
variable

```
        count--;
```

```
        txt.setText(""+count);
```

Set text
on screen

```
    }
```

```
    public void addtoit(View view) {
```

```
        TextView txt = (TextView) findViewById(R.id.numberView);
```

```
        count++;
```

```
        txt.setText(""+count);
```

```
    }
```

```
}
```

Fill in the XML and
Java for this button



Method name: openNow

Button id: open

White on Blue

Text Size: 40 sp

XML :

<Button

```
    android:id="@+id/_____"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="_____"
    android:text="_____"
    android:background="_____"
    android:textColor="_____"
    android:textSize="_____" />
```

MainActivity:

```
    public void _____(View view){
    }
```

Fill in the XML and
Java for this button

Done

Method name: Done
Button id: DoneButton
White on Black
Text Size: 60 sp

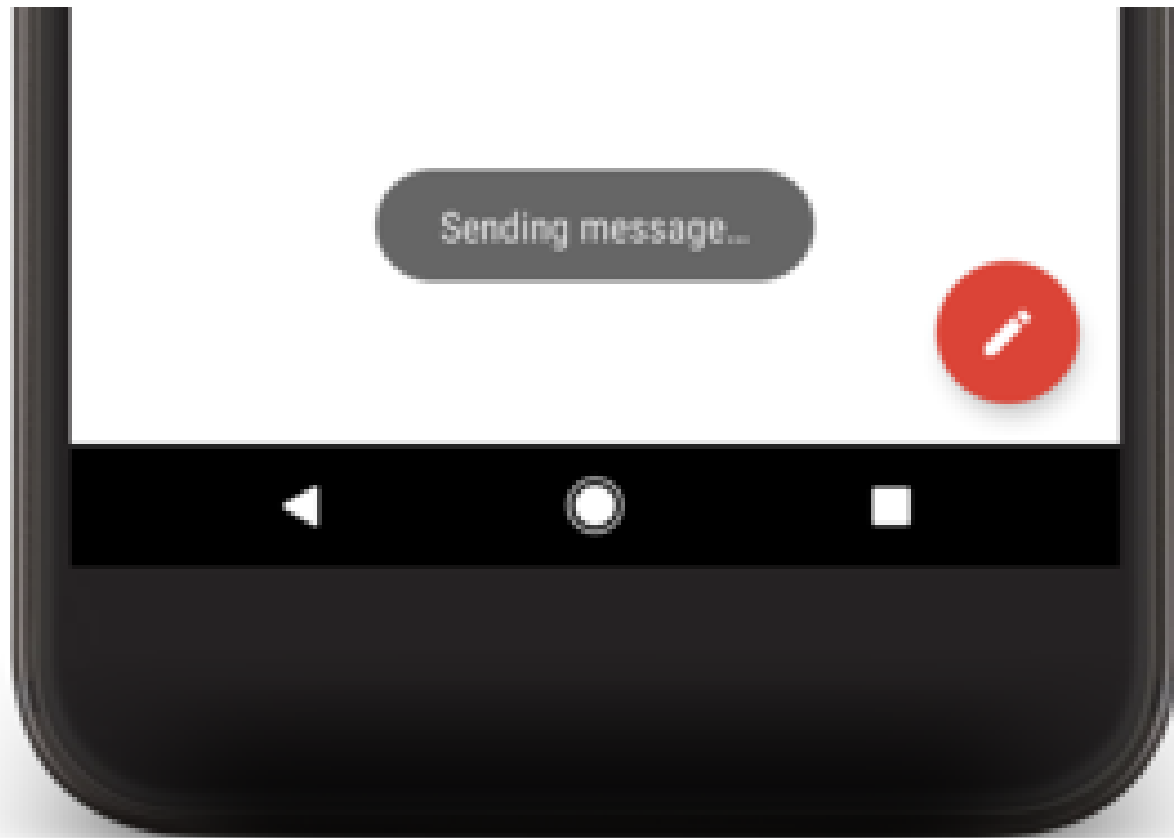
XML :

<Button

```
    android:id="@+id/_____"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:onClick="_____"  
    android:text="_____"  
    android:background="_____"  
    android:textColor="_____"  
    android:textSize="_____" />
```

MainActivity:

```
    public void _____(View view){  
    }
```



A toast provides simple feedback about an operation in a small popup.

It only fills the amount of space required for the message and the current activity remains visible and interactive.

Toasts automatically disappear after a timeout.

A gray rounded rectangle with the word "Hello" in white text.

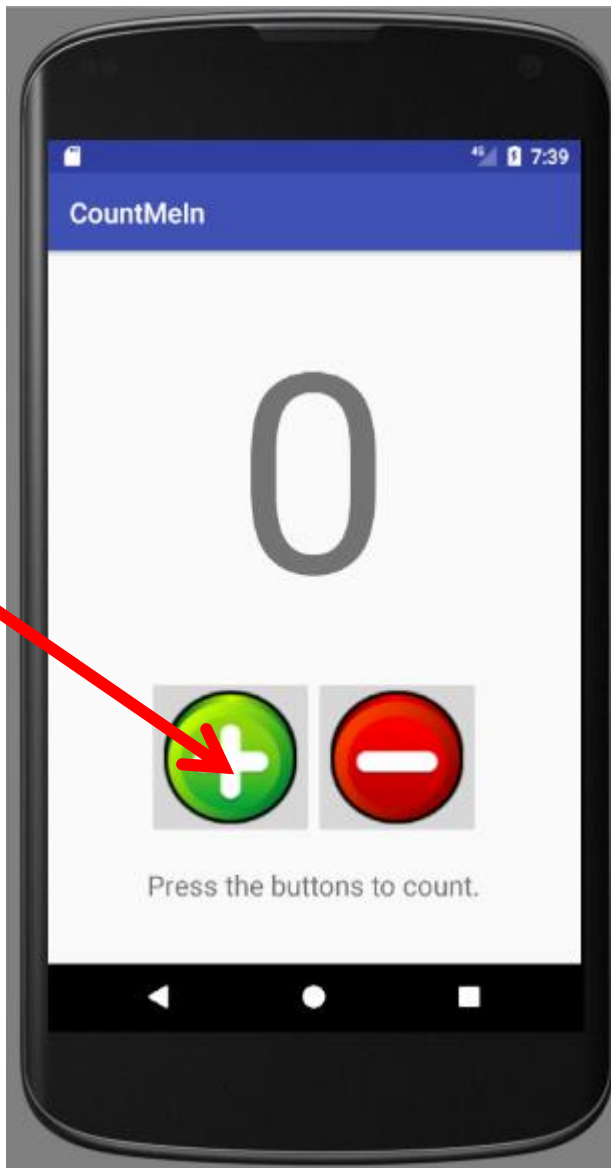
Fill in the toast's code with the text shown.

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


Saving Choice

Fill in the toast's code with the text shown.

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



```
public class MainActivity extends AppCompatActivity {  
    int count = 0;  
    @Override  
    protected void onCreate(Bundle savedInstanceState)  
    {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

```
public void subtract(View view){  
    TextView txt = (TextView) findViewById(R.id.numberView);  
    count--;  
    txt.setText(""+count);  
}
```

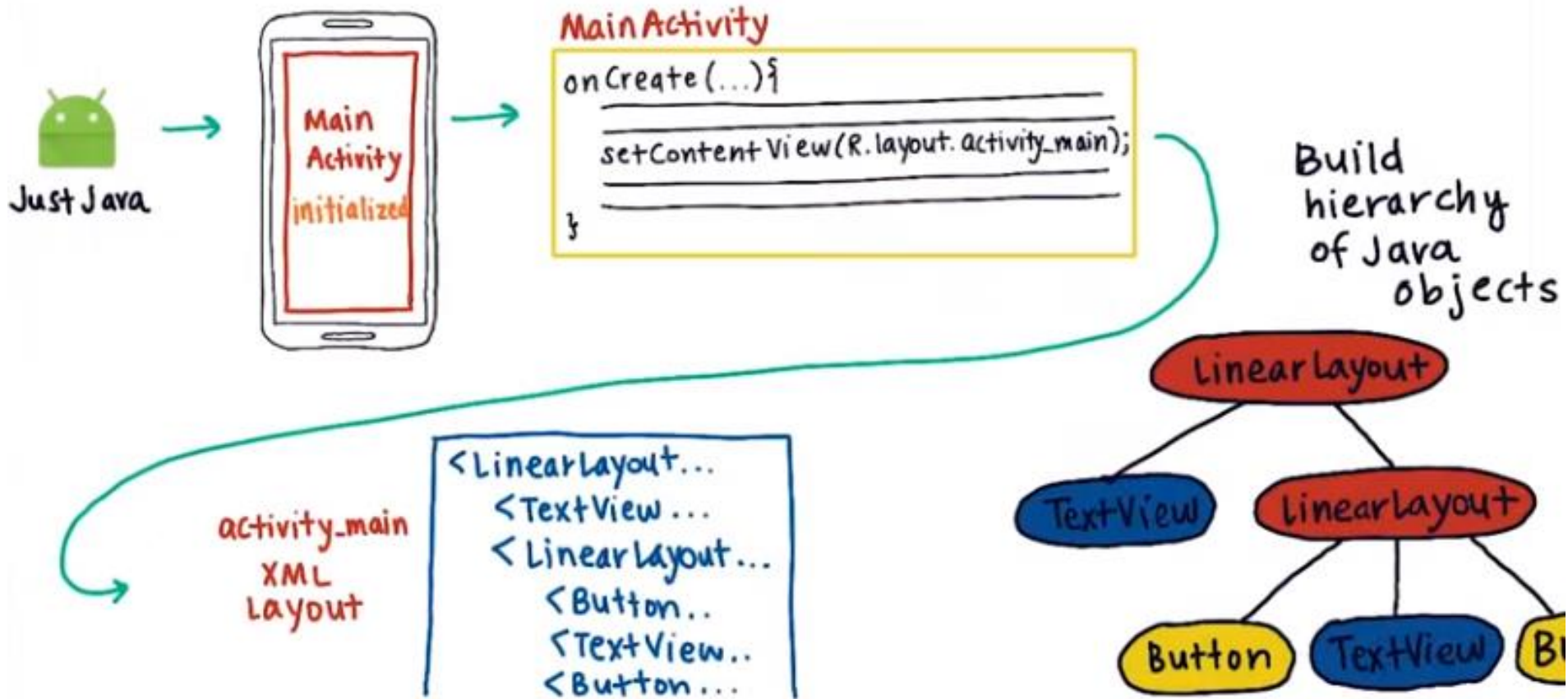
```
public void addtoit(View view){  
    TextView txt = (TextView) findViewById(R.id.numberView);  
    count++;  
    txt.setText(""+count);  
}
```

```
}
```



Main
Activity

WHAT'S HAPPENING BETWEEN XML & JAVA



onCreate = init.

onCreate goes through the XML file and writes the java code for it.
Called "inflation" process.

```
TextView txt=(TextView) findViewById(R.id.numberView);
```

Java variable name

XML variable name

```
ImageView dice = (ImageView) findViewById(R.id.dice);
```

The java variable name and the XML variable name can be the same.

Fill in these pieces of inflation code.

```
_____ response= (TextView) findViewById (R.id. response) ;
```

```
ImageView pic = (ImageView) _____ (R.id. pic) ;
```

```
Button click = ( _____ ) findViewById (R.id. click) ;
```

Inflation

- The inflation process occurs in the java code in the onCreate method
- Inflation is the process of translating the XML file into Java so that it can be coded.
- This is **good** because it allows us to design in XML – which is easier – and to code in Java – which actually is a full programming language.
- This is **bad** because during the inflation process, all of the id's get lost.
- After inflation, if you wish to use them, you have the findViewById in the java code to use them.

OnClick

- An attribute added to buttons and imageViews that allows it to be clicked by a user.
- To make it work, you make a method with the EXACT name of the onClick inside the java file:
- `public void _____ (View view) { }`

id

- An attribute added to things that need to change.
- Inside the Java file, because of inflation, you will need to use the `findViewById` to look it up.
- All of the ids for the program are stored in a file named “R”.

<ImageView

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/kittycat"  
    android:scaleType="center"  
    android:onClick="cat"  
    android:id="@+id/meow" />
```

Working
Buttons

1. Add the onClick
to the View

2. Make a method with
the onClick name and
the parameter View view

```
public void cat (View view) {
```

```
    Toast.makeText(getApplicationContext(), "mew mew", Toast.LENGTH_SHORT).show();  
}
```

<ImageView

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/kittycat"  
    android:scaleType="center"  
    android:onClick="cat"  
    android:id="@+id/meow" />
```

Changeable
ImageViews

1. Add an id to the view

```
public void cat (View view) {
```

```
    ImageView meow = (ImageView) findViewById(R.id.meow) ;  
    meow.setImageResource(R.drawable.giantCat) ;
```

```
}
```

2. Find the view

3. Change the view as
needed

<TextView

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="kittycat"  
    android:onClick="cat"  
    android:id="@+id/meow" />
```

Changeable
TextViews

1. Add an id to the view

```
public void cat (View view) {
```

```
    TextView meow = (TextView) findViewById(R.id.meow) ;  
    meow.setText("I like tuna") ;
```

```
}
```

2. Find the view

3. Change the view as needed

ca gorskicompsci www buttontests MainActivity

activity_main.xml x MainActivity.java x

MainActivity clicked()

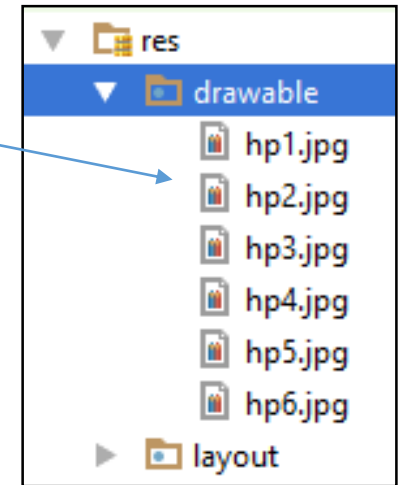
```
1 package ca.gorskicompsci.www.buttontests;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13
14     public void clicked(View view) {
15
16     }
17 }
18
```

Pretty much,
You always
want to add
whatever they
suggest.

? android.view.View? Alt+Enter

Some code snippets:

```
public void clicked(View view) {  
    TextView txt = (TextView) findViewById(R.id.txt) ;  
    ImageView pic = (ImageView) findViewById(R.id.pic) ;  
    EditText write = (EditText) findViewById(R.id.write) ;  
  
    String word = write.getText().toString() ;  
    txt.setText(word) ;  
    pic.setImageResource(R.drawable.hp2) ;  
    pic.setEnabled(false) ;  
  
    int num = Integer.parseInt(word) ;  
    int rNum = (int) (Math.random() * 3 + 1) ;  
  
    txt.setText(num + " " + rNum) ;  
  
}
```



What goes in blank A?



stego.png



brono.png

In XML:

```
<ImageView
    android:id="@+id/dino"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:onClick="dinoClick"
    android:src="@drawable/_____A" />
```

In MainActivity.java:

```
public void _____B (View view) {
    ImageView _____C = (ImageView) findViewById(R.id. _____D);
    _____E.setImageResource(R.drawable. _____F);
}
```

What goes in blank B?



stego.png



brono.png

In XML:

```
<ImageView
    android:id="@+id/dino"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:onClick="dinoClick"
    android:src="@drawable/_____A" />
```

In MainActivity.java:

```
public void _____B (View view) {
    ImageView _____C = (ImageView) findViewById(R.id. _____D);
    _____E.setImageResource(R.drawable. _____F);
}
```

What goes in
blank C?



stego.png



brono.png

In XML:

```
<ImageView
    android:id="@+id/dino"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:onClick="dinoClick"
    android:src="@drawable/_____A" />
```

In MainActivity.java:

```
public void _____B (View view) {
    ImageView _____C = (ImageView) findViewById(R.id. _____D);
    _____E.setImageResource(R.drawable. _____F);
}
```


What goes in blank D?



stego.png



brono.png

In XML:

```
<ImageView
    android:id="@+id/dino"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:onClick="dinoClick"
    android:src="@drawable/_____A" />
```

In MainActivity.java:

```
public void _____B (View view) {
    ImageView _____C = (ImageView) findViewById(R.id. _____D);
    _____E.setImageResource(R.drawable. _____F);
}
```

What goes in blank E?



stego.png



brono.png

In XML:

```
<ImageView
    android:id="@+id/dino"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:onClick="dinoClick"
    android:src="@drawable/_____A" />
```

In MainActivity.java:

```
public void _____B (View view) {
    ImageView _____C = (ImageView) findViewById(R.id. _____D);
    _____E.setImageResource(R.drawable. _____F);
}
```

What goes in blank F?



stego.png



brono.png

In XML:

```
<ImageView
    android:id="@+id/dino"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:onClick="dinoClick"
    android:src="@drawable/_____A" />
```

In MainActivity.java:

```
public void _____B (View view) {
    ImageView _____C = (ImageView) findViewById(R.id. _____D);
    _____E.setImageResource(R.drawable. _____F);
}
```



stego.png



brono.png

In XML:

```
<ImageView
    android:id="@+id/dino"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:onClick="dinoClick"
    android:src="@drawable/_____A stego" />
```

In MainActivity.java:

```
public void _____B dinoClick (View view) {
    ImageView _____C dino = (ImageView) findViewById(R.id. _____D dino);
    _____E dino.setImageResource(R.drawable. _____F brono);
}
```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="center_horizontal"
    android:orientation="vertical">
    <ImageView
        android:id="@+id/swimmer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="switchMe"
        android:src="@drawable/blue"
        android:padding="100dp"/>

</LinearLayout>

```

```

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void switchMe ( ) {
        int n = (int) (Math.random() * 4);
        ImageView fish = ( ) findViewById(R.id. swimmer );
        if(n==1)
            fish.setImageResource(R.drawable. );
        else if(n==2)
            . (R.drawable.red);
        else if(n==3)
            .setImageResource(R. .green);
        else
            .setImageResource( .drawable.yellow);
    }
}

```

