Week-2 Homework

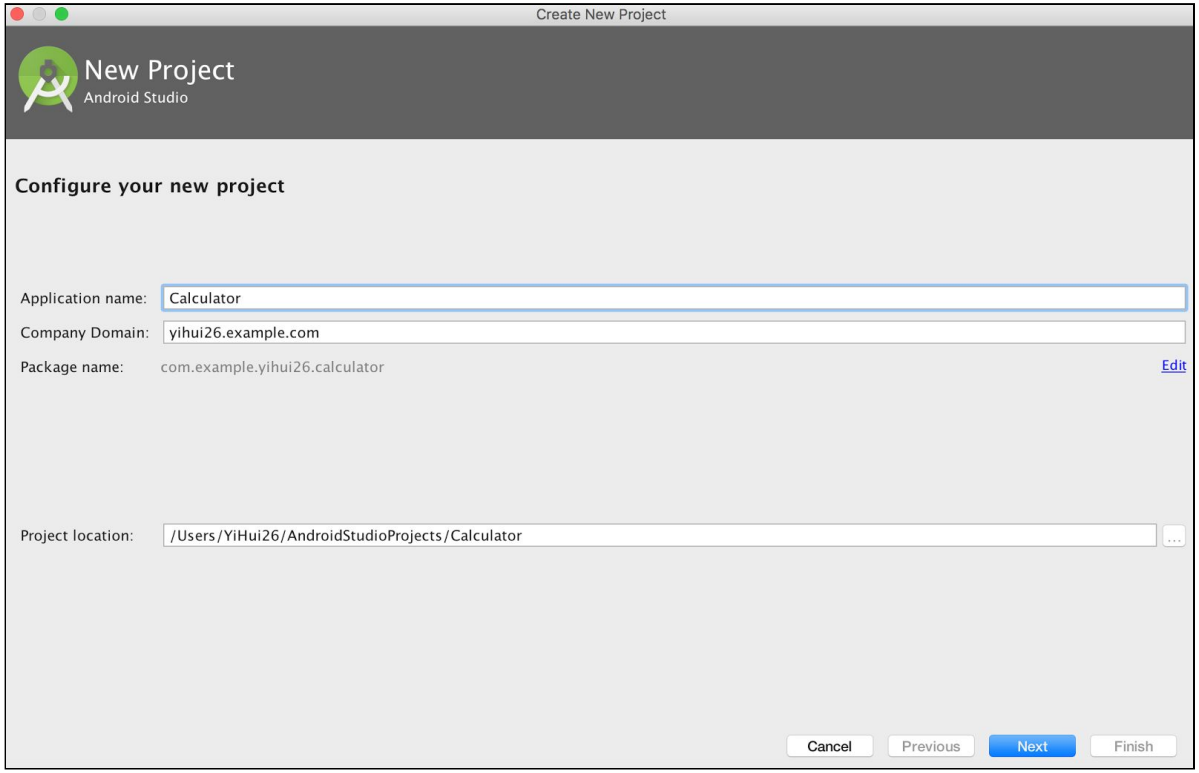
Elaine (Yi Hui) Aw

Write an Android or iPhone Calculator application:

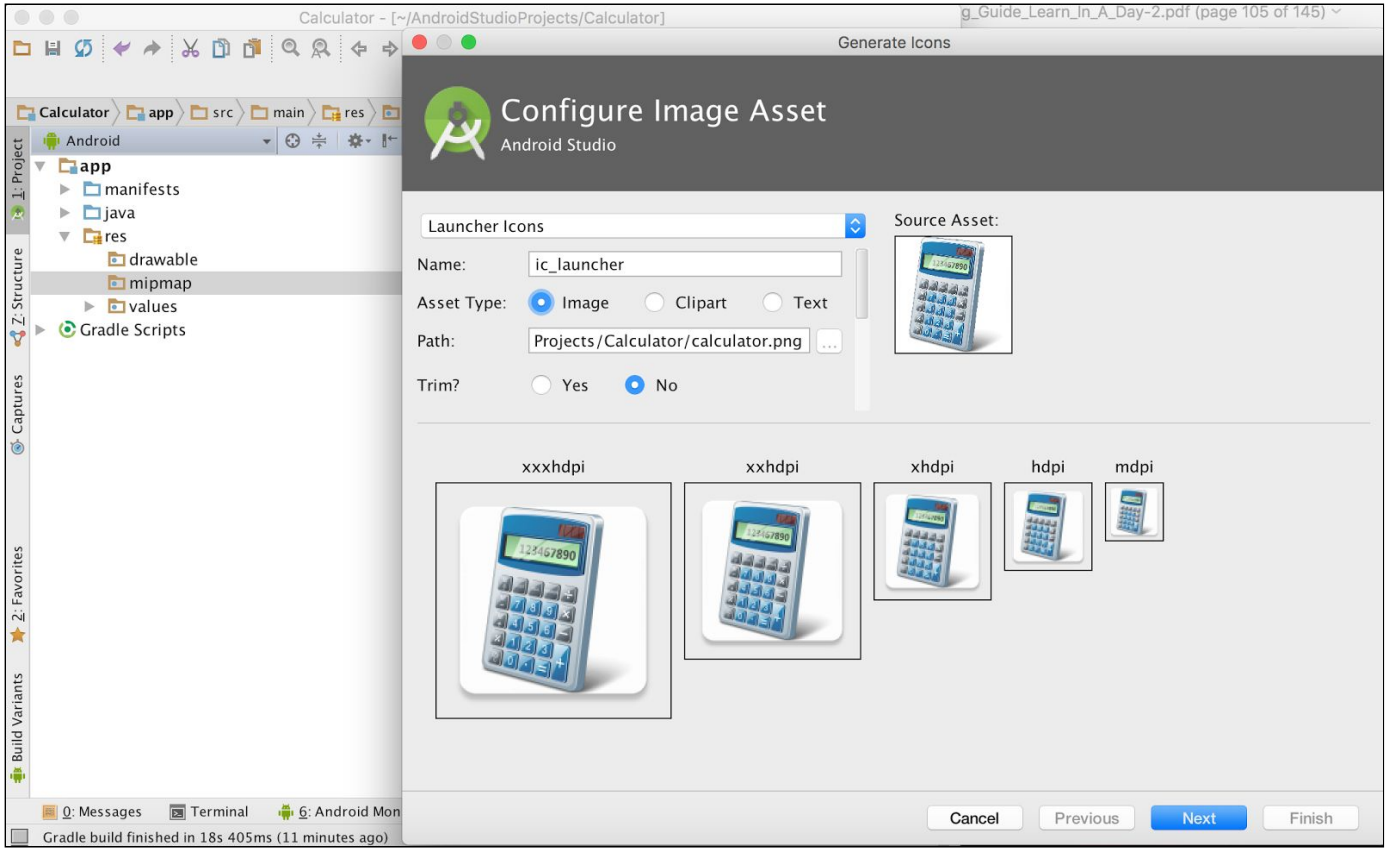
- Chosen to develop the application on Android because I have Java background, and I can follow the textbook sample.

Android (With reference to Chapter 20)

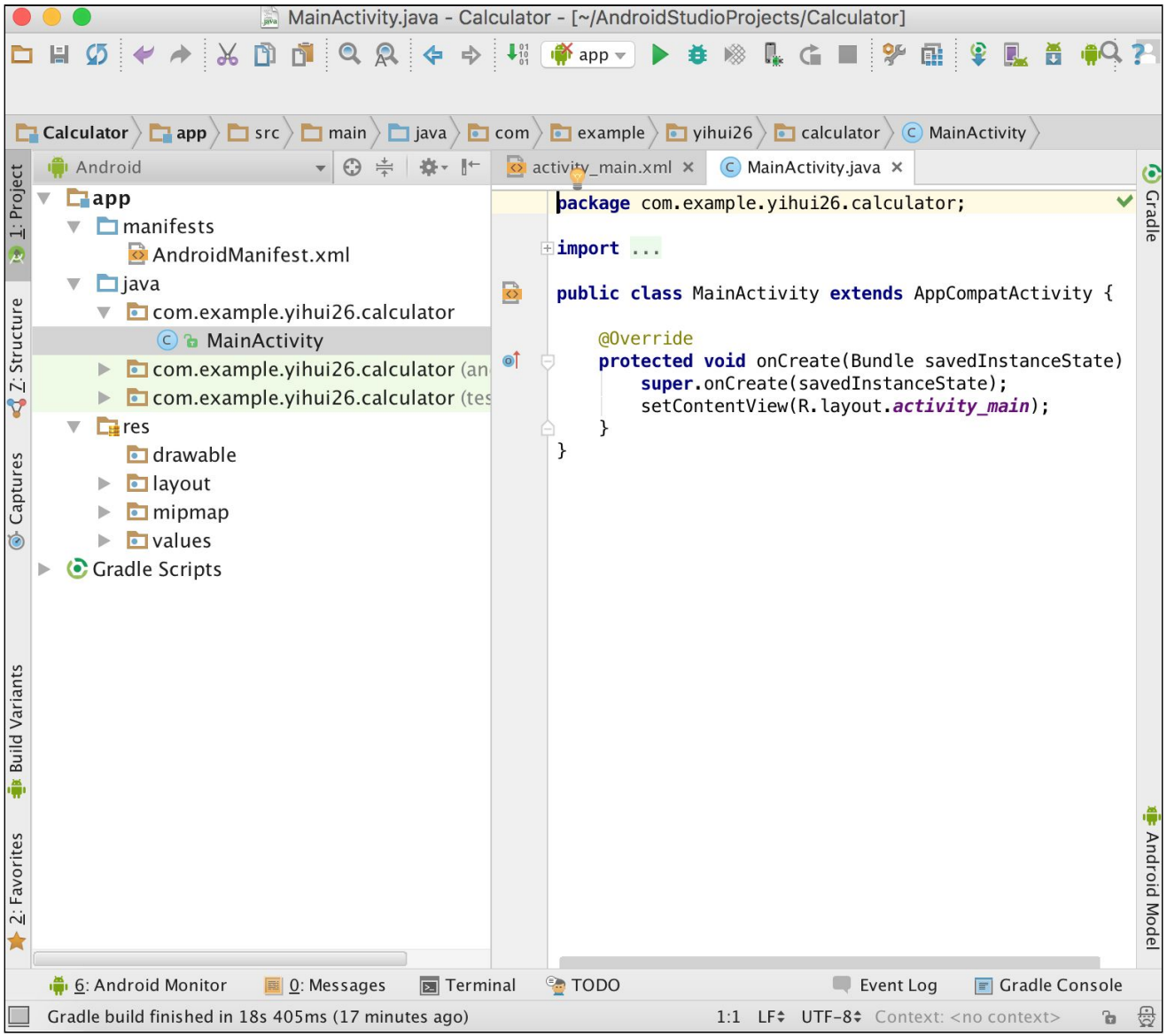
Step 1: New application named “Calculator”.

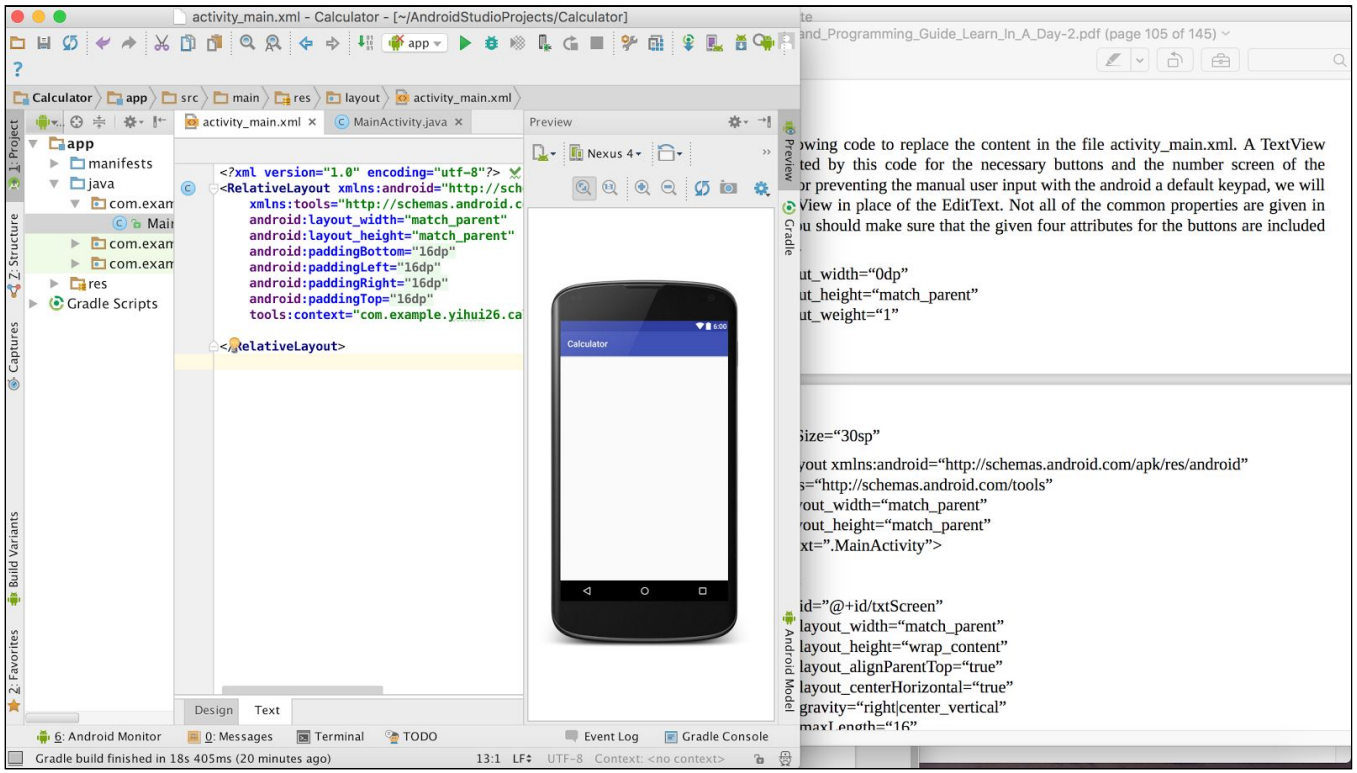


Step 2-5: Replace default Green Robot icon with a Calculator Icon.

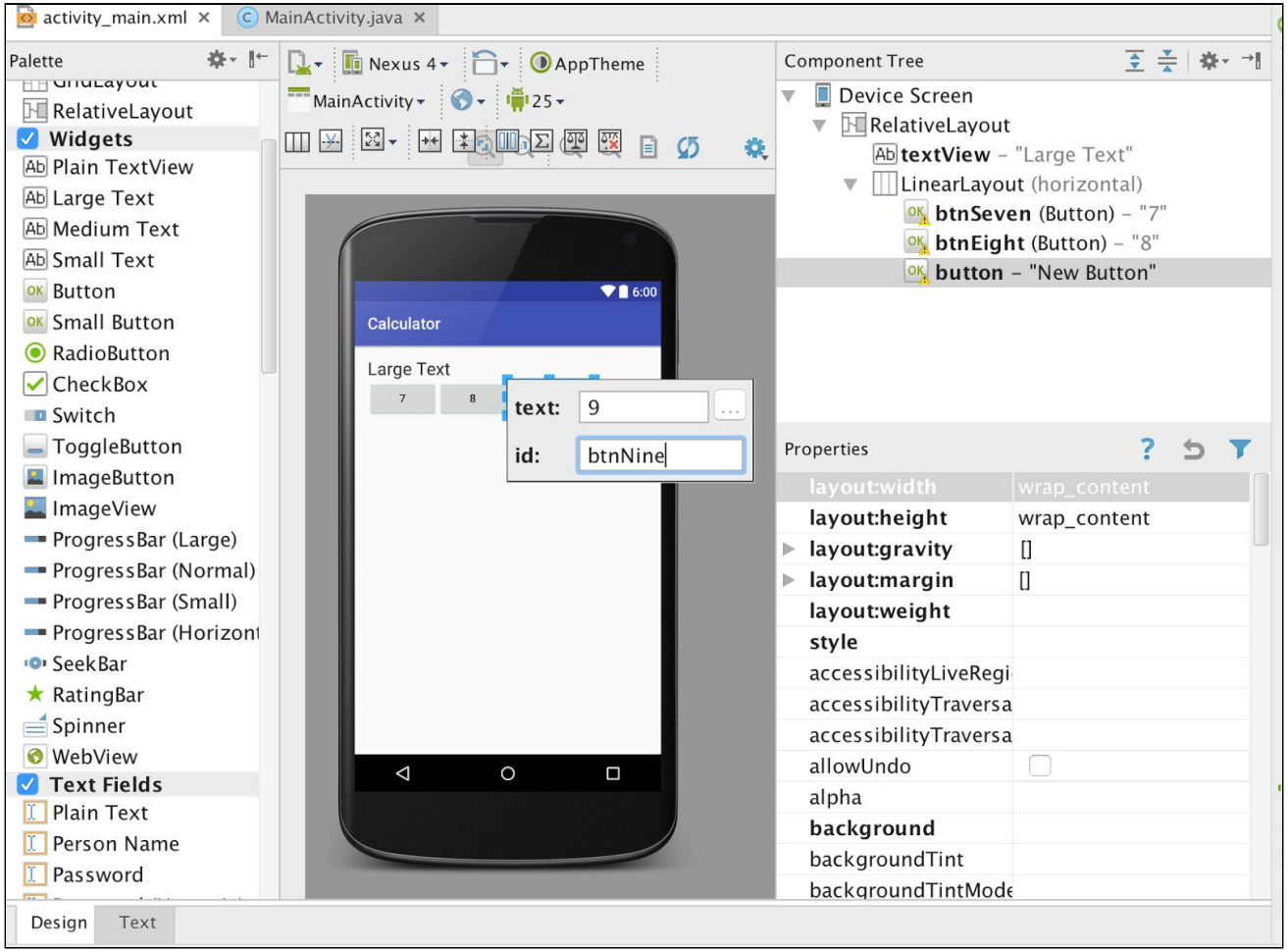


Step 6: Create new blank activity and update the xml file.

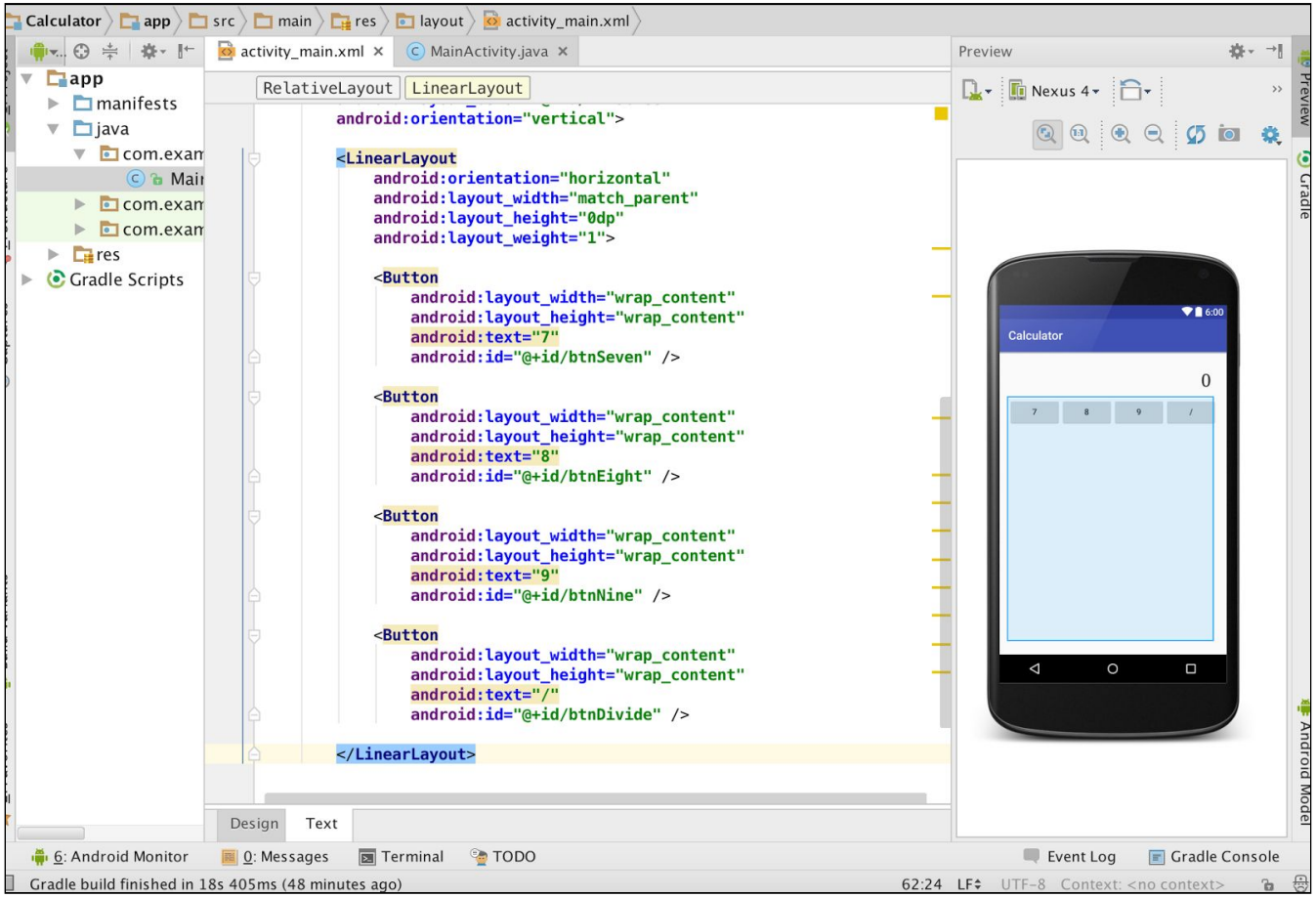

MainActivity.java - Calculator - [~/AndroidStudioProjects/Calculator]
app
Calculator
app
src
main
java
com
example
yihui26
calculator
MainActivity
Android
activity_main.xml
MainActivity.java
app
manifests
AndroidManifest.xml
java
com.example.yihui26.calculator
MainActivity
com.example.yihui26.calculator (an
com.example.yihui26.calculator (tes
res
drawable
layout
mipmap
values
Gradle Scripts
package com.example.yihui26.calculator;
import ...
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState)
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
}
1: Project
7: Structure
Captures
Build Variants
2: Favorites
Gradle
Android Model
6: Android Monitor
0: Messages
Terminal
TODO
Event Log
Gradle Console
Gradle build finished in 18s 405ms (17 minutes ago)
1:1
LF
UTF-8
Context: <no context>



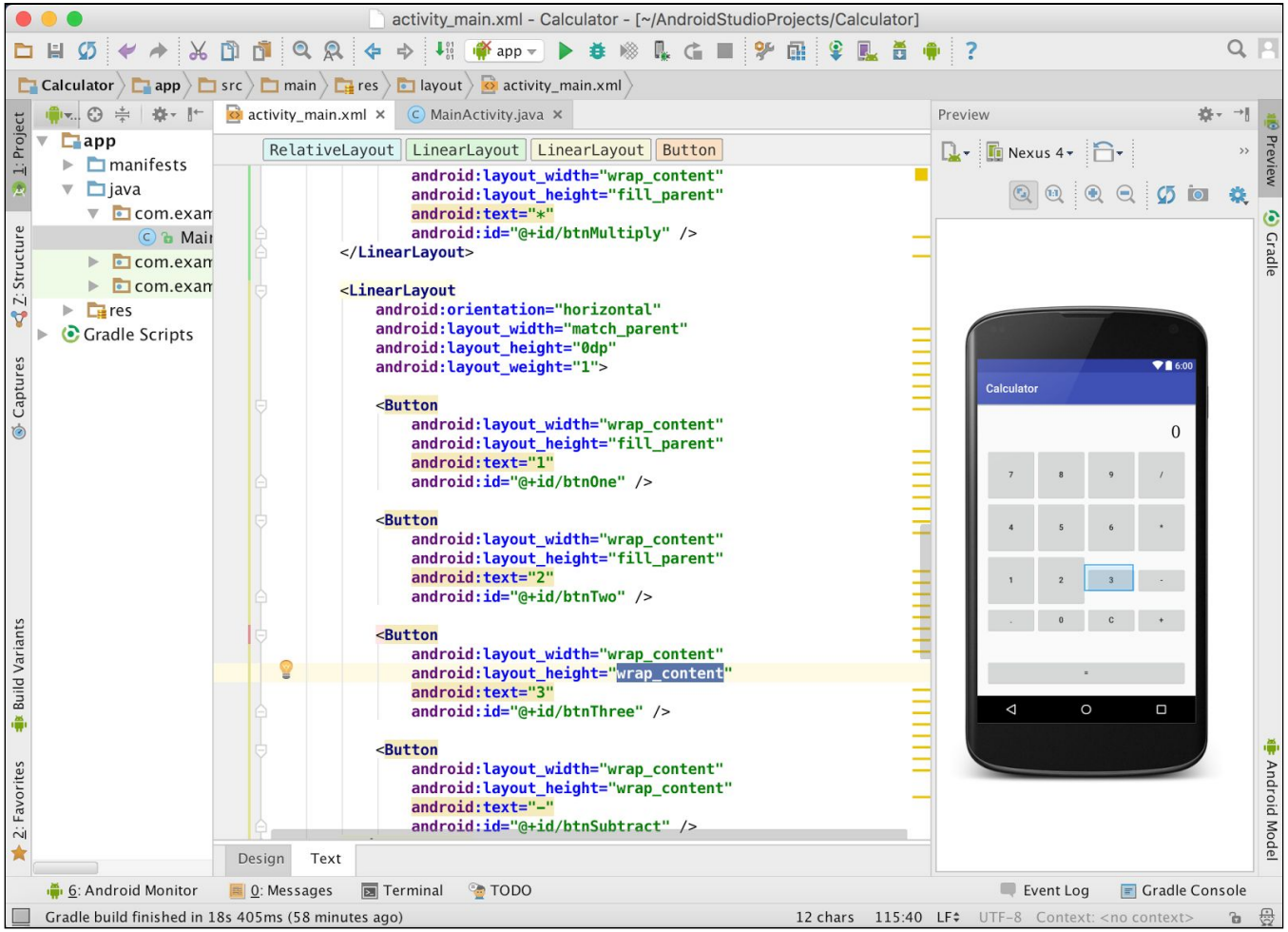
Code seems to be not working -- decided to build directly from the viewer:



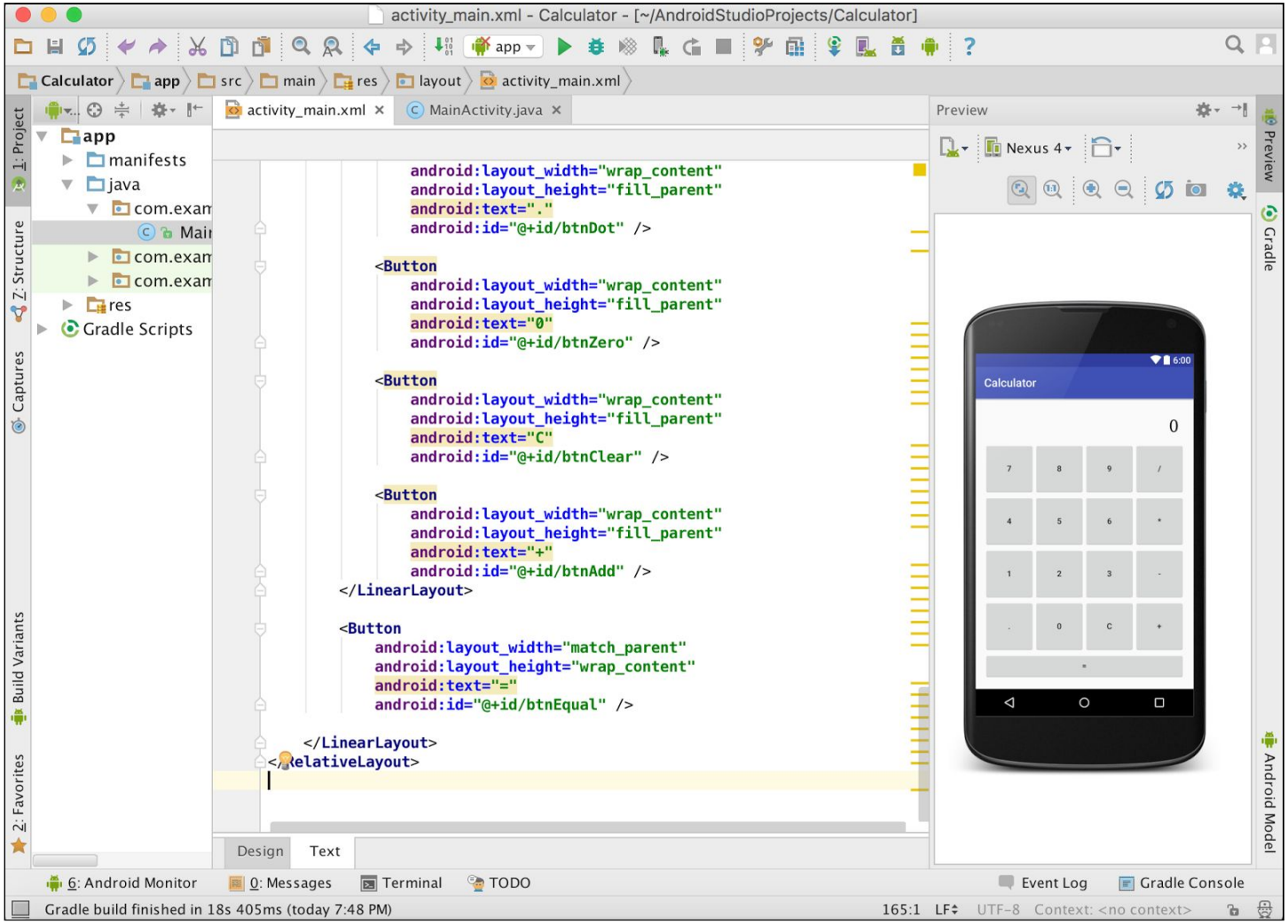
Generated xml can be used to populate the other rows of the calculator easily --



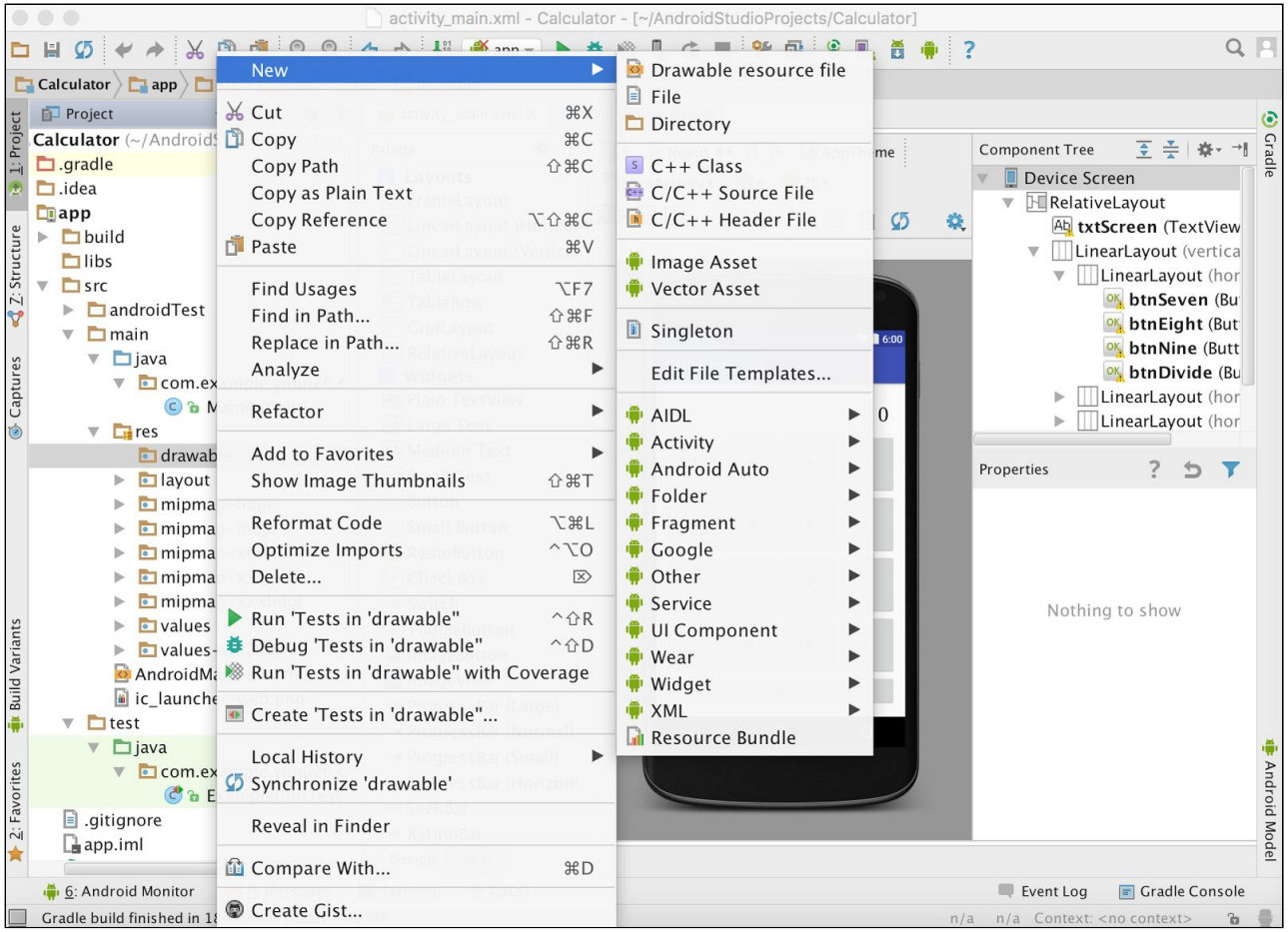
Align the buttons


activity_main.xml - Calculator - [~/AndroidStudioProjects/Calculator]
Calculator
app
src
main
res
layout
activity_main.xml
activity_main.xml
MainActivity.java
RelativeLayout
LinearLayout
LinearLayout
Button
android:layout_width="wrap_content"
android:layout_height="fill_parent"
android:text="*"
android:id="@+id/btnMultiply" />
</LinearLayout>
<LinearLayout
android:orientation="horizontal"
android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_weight="1">
<Button
android:layout_width="wrap_content"
android:layout_height="fill_parent"
android:text="1"
android:id="@+id/btnOne" />
<Button
android:layout_width="wrap_content"
android:layout_height="fill_parent"
android:text="2"
android:id="@+id/btnTwo" />
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="3"
android:id="@+id/btnThree" />
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="-"
android:id="@+id/btnSubtract" />
Preview
Nexus 4
Calculator
0
Design
Text
6: Android Monitor
0: Messages
Terminal
TODO
Event Log
Gradle Console
Gradle build finished in 18s 405ms (58 minutes ago)
12 chars
115:40
LF
UTF-8
Context: <no context>

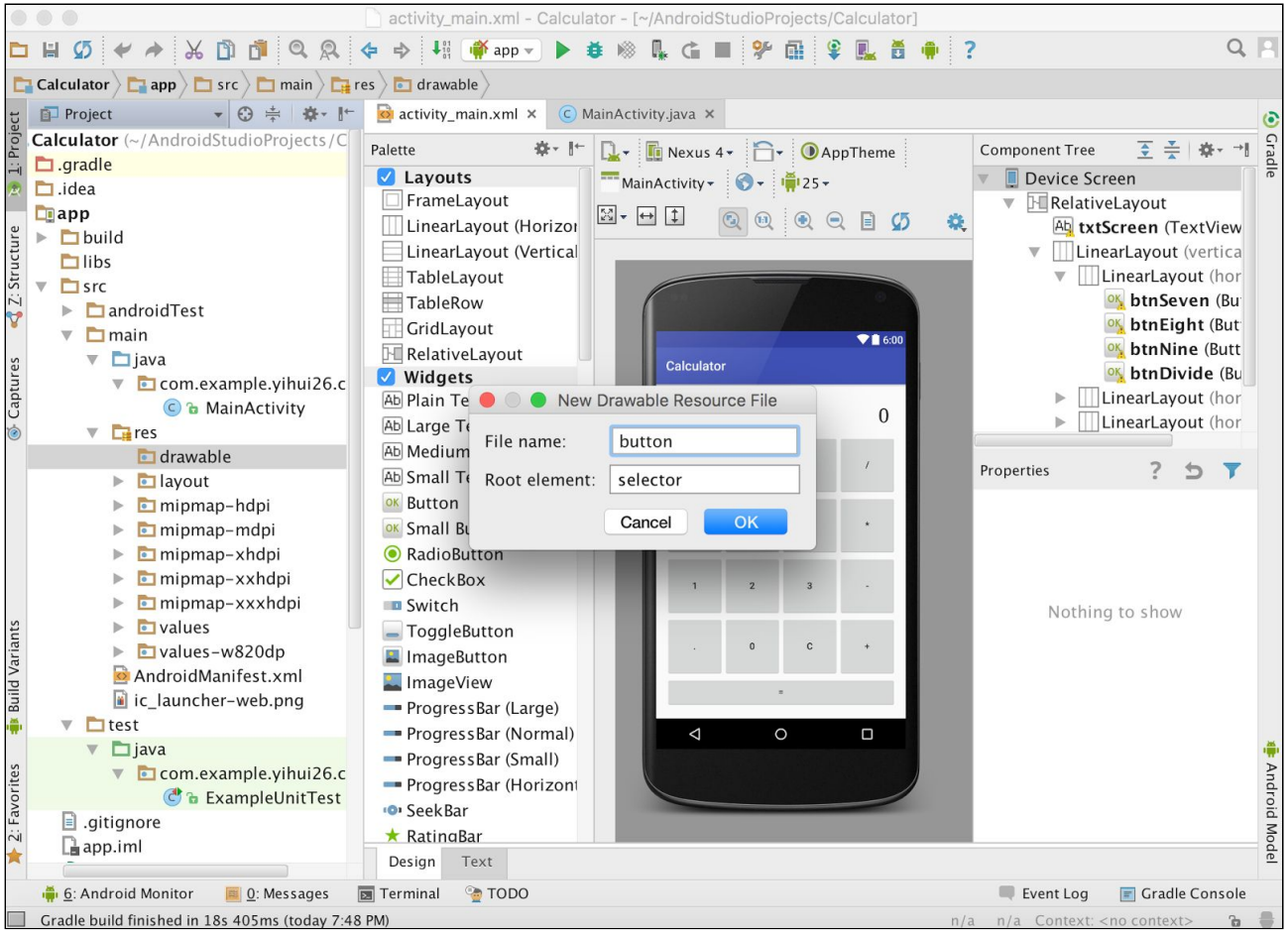
Done


activity_main.xml - Calculator - [~/AndroidStudioProjects/Calculator]
app
Calculator
app
src
main
res
layout
activity_main.xml
activity_main.xml
MainActivity.java
android:layout_width="wrap_content"
android:layout_height="fill_parent"
android:text="."
android:id="@+id/btnDot" />
<Button
android:layout_width="wrap_content"
android:layout_height="fill_parent"
android:text="0"
android:id="@+id/btnZero" />
<Button
android:layout_width="wrap_content"
android:layout_height="fill_parent"
android:text="C"
android:id="@+id/btnClear" />
<Button
android:layout_width="wrap_content"
android:layout_height="fill_parent"
android:text="+"
android:id="@+id/btnAdd" />
</LinearLayout>
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="="
android:id="@+id/btnEqual" />
</LinearLayout>
</RelativeLayout>
Design
Text
Preview
Nexus 4
Calculator
0
1: Project
7: Structure
Captures
Build Variants
2: Favorites
Gradle
Android Model
6: Android Monitor
0: Messages
Terminal
TODO
Event Log
Gradle Console
Gradle build finished in 18s 405ms (today 7:48 PM)
165:1
LF
UTF-8
Context: <no context>

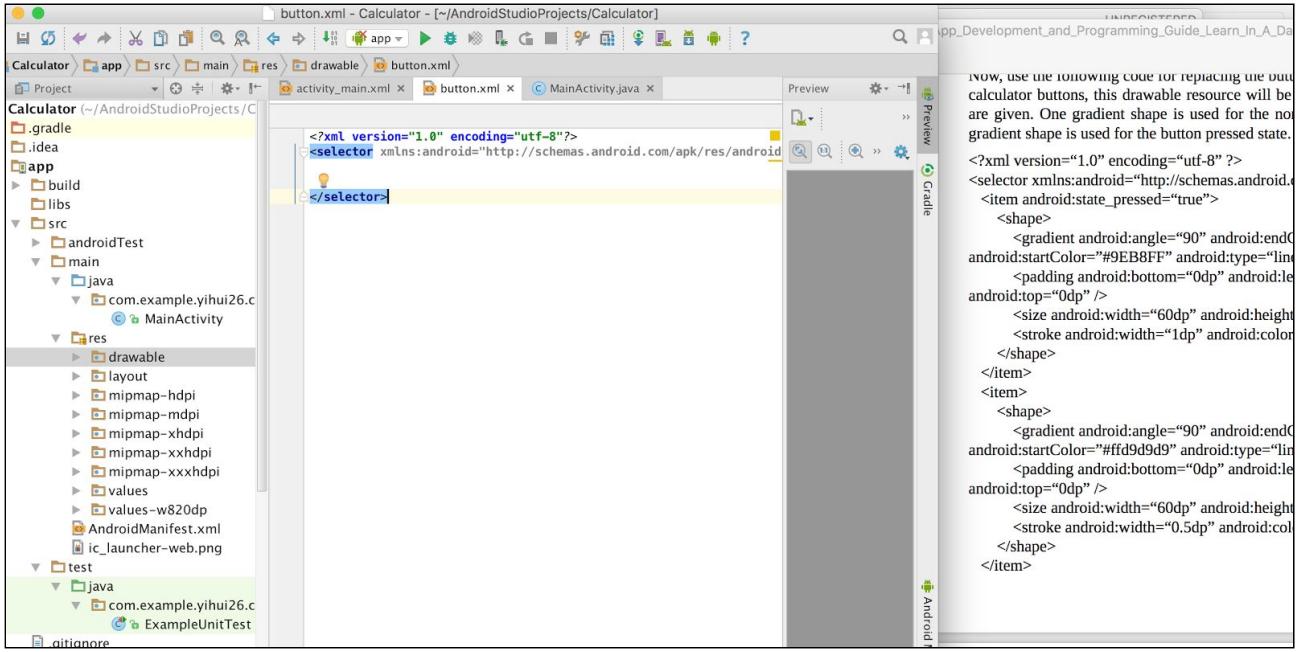
Step 7: Create new drawable resource file

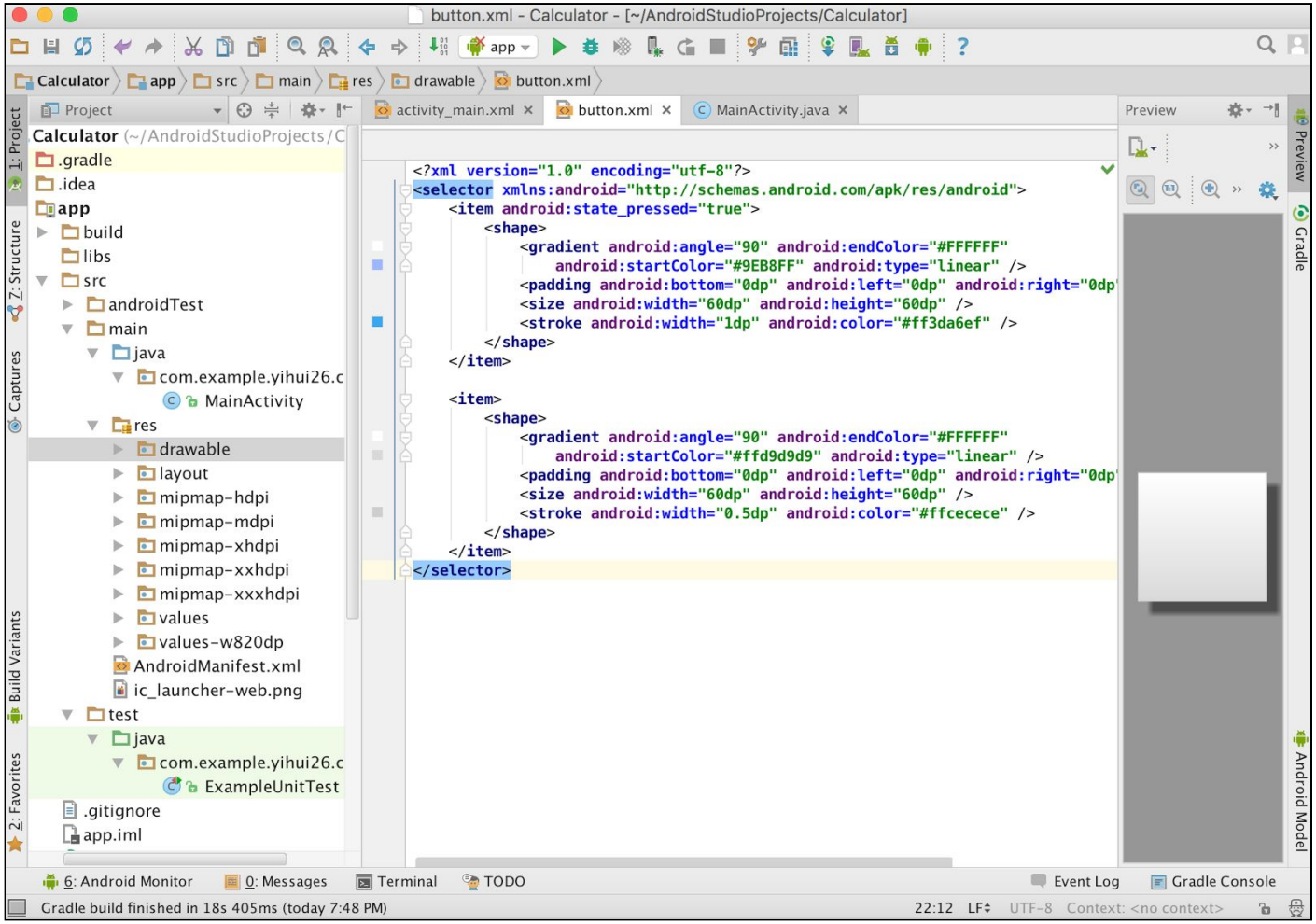


Step 8: Create new drawable file named “button”.

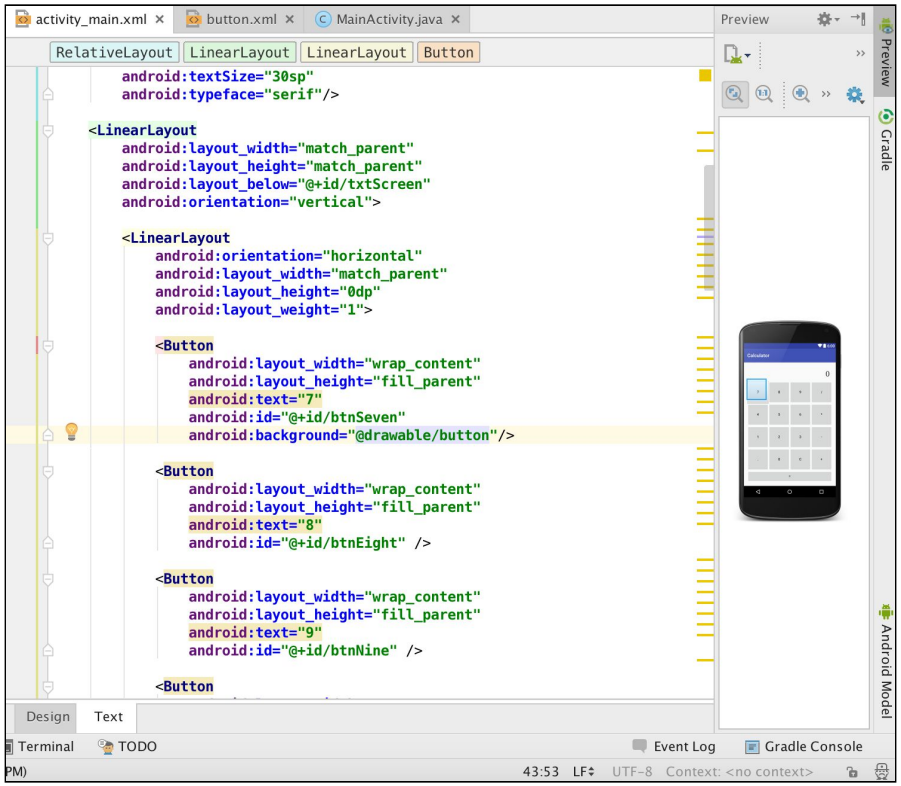


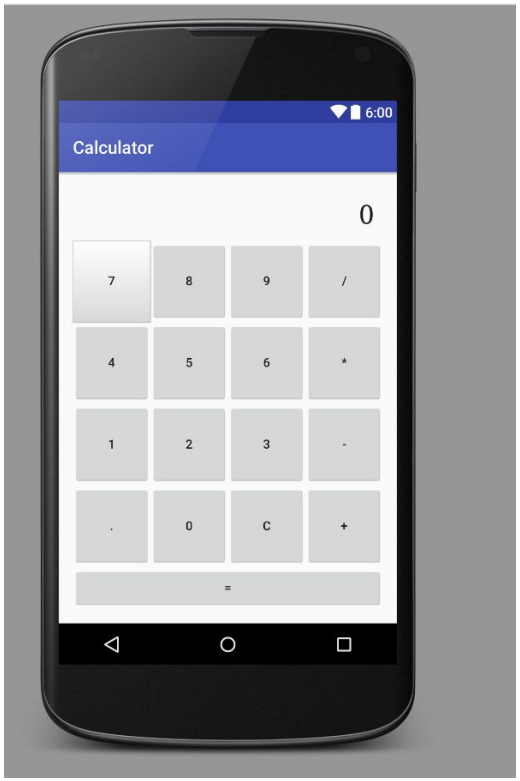
Step 9: Fill up button.xml properties as the book.



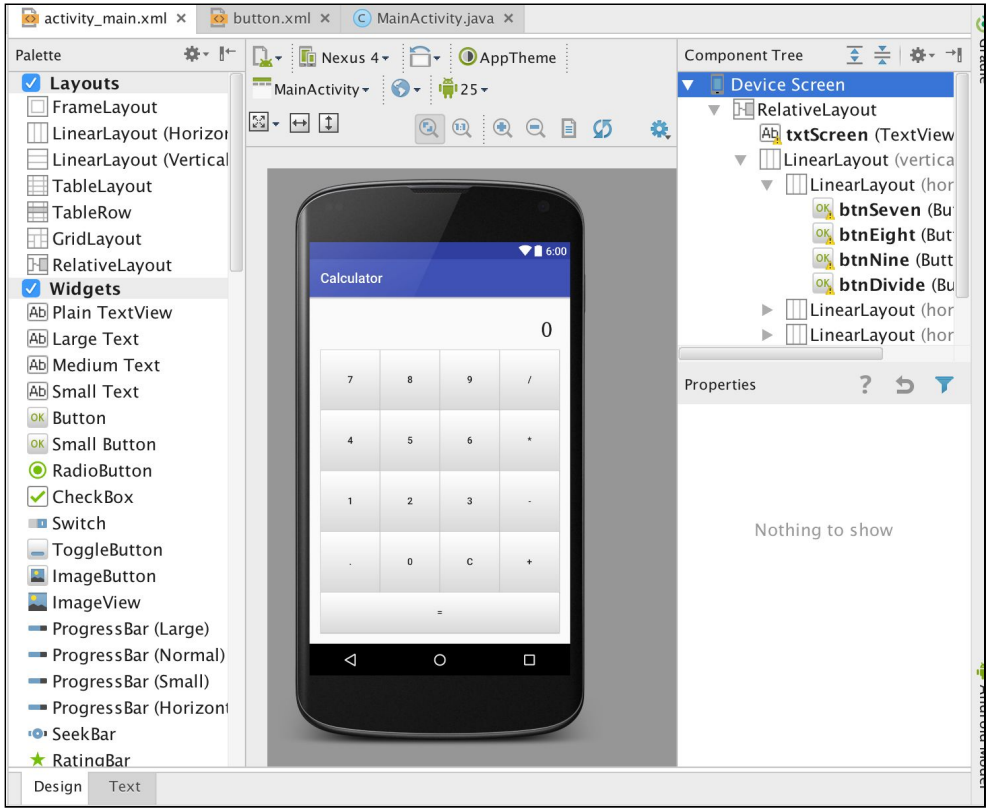

button.xml - Calculator - [~/AndroidStudioProjects/Calculator]
app
Calculator
app
src
main
res
drawable
button.xml
Project
activity_main.xml
button.xml
MainActivity.java
Preview
Calculator (~/AndroidStudioProjects/C
.gradle
.idea
app
build
libs
src
androidTest
main
java
com.example.yihui26.c
MainActivity
res
drawable
layout
mipmap-hdpi
mipmap-mdpi
mipmap-xhdpi
mipmap-xxhdpi
mipmap-xxxhdpi
values
values-w820dp
AndroidManifest.xml
ic_launcher-web.png
test
java
com.example.yihui26.c
ExampleUnitTest
.gitignore
app.iml
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
<item android:state_pressed="true">
<shape>
<gradient android:angle="90" android:endColor="#FFFFFF"
android:startColor="#9EB8FF" android:type="linear" />
<padding android:bottom="0dp" android:left="0dp" android:right="0dp
<size android:width="60dp" android:height="60dp" />
<stroke android:width="1dp" android:color="#ff3da6ef" />
</shape>
</item>
<item>
<shape>
<gradient android:angle="90" android:endColor="#FFFFFF"
android:startColor="#ffd9d9d9" android:type="linear" />
<padding android:bottom="0dp" android:left="0dp" android:right="0dp
<size android:width="60dp" android:height="60dp" />
<stroke android:width="0.5dp" android:color="#ffcecece" />
</shape>
</item>
</selector>
1: Project
7: Structure
Captures
Build Variants
2: Favorites
Preview
Gradle
Android Model
6: Android Monitor
0: Messages
Terminal
TODO
Event Log
Gradle Console
Gradle build finished in 18s 405ms (today 7:48 PM)
22:12
LF
UTF-8
Context: <no context>

Step 10: Add property “android:background” to all activity_main.xml buttons.

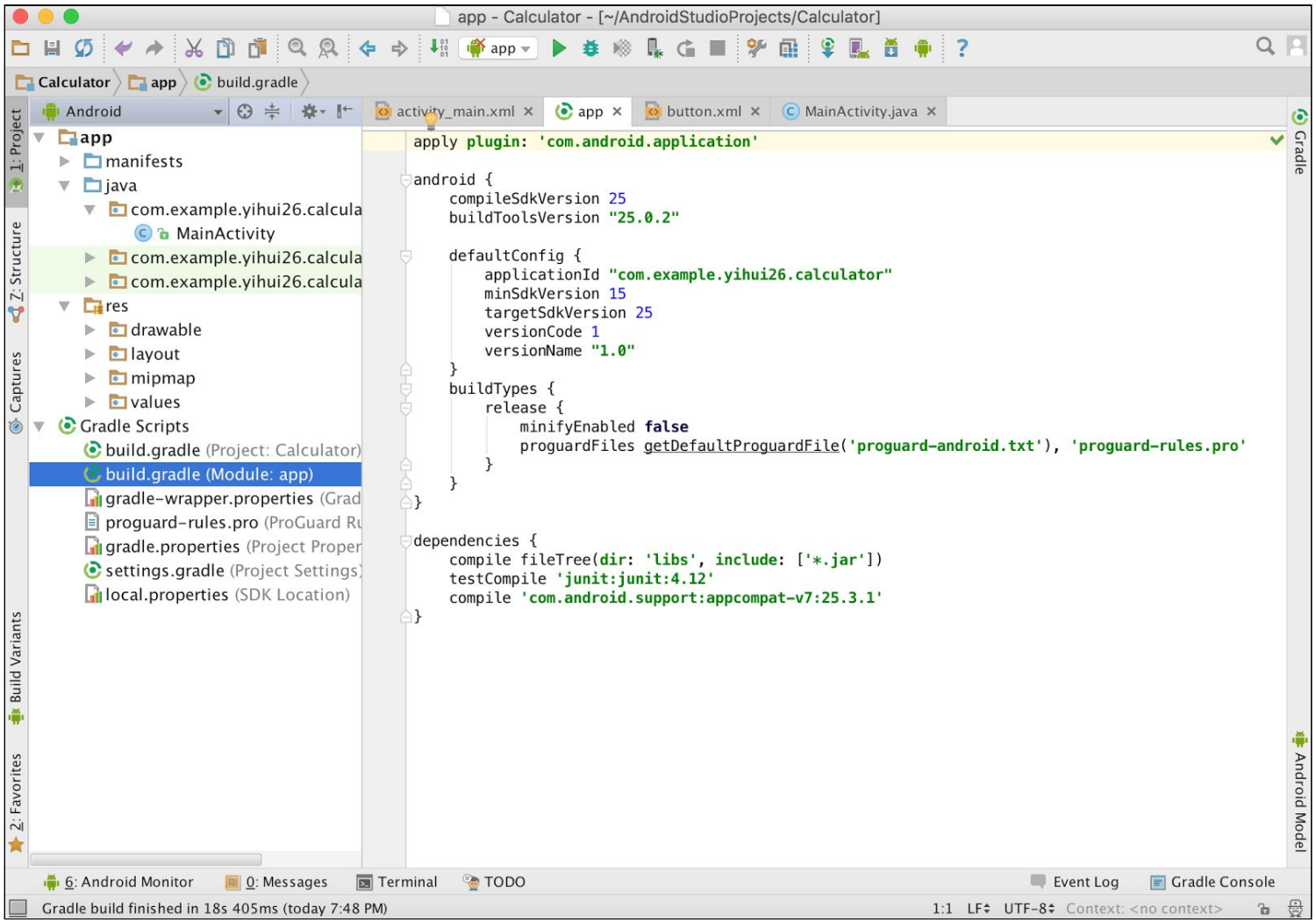


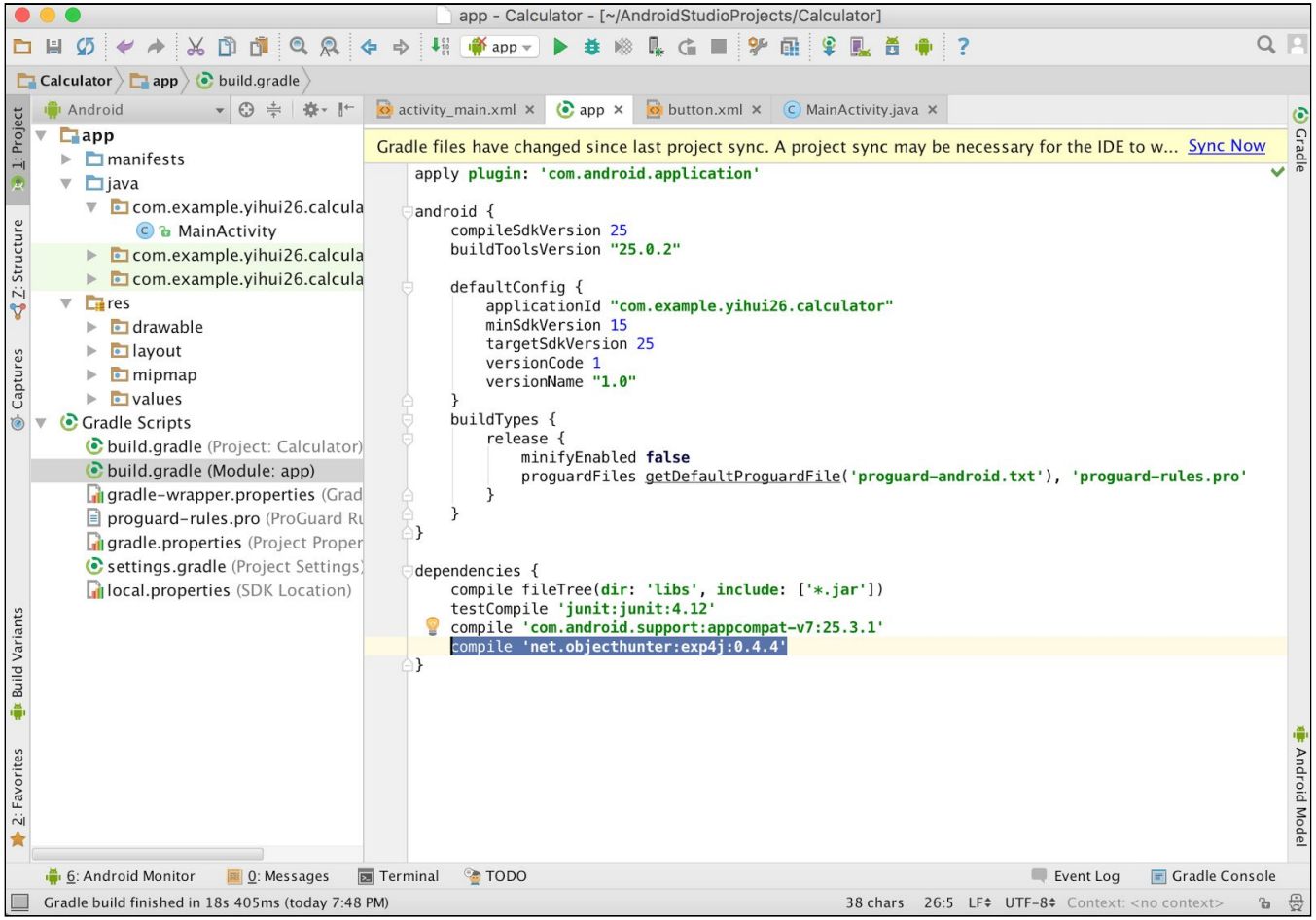


After updating all buttons --

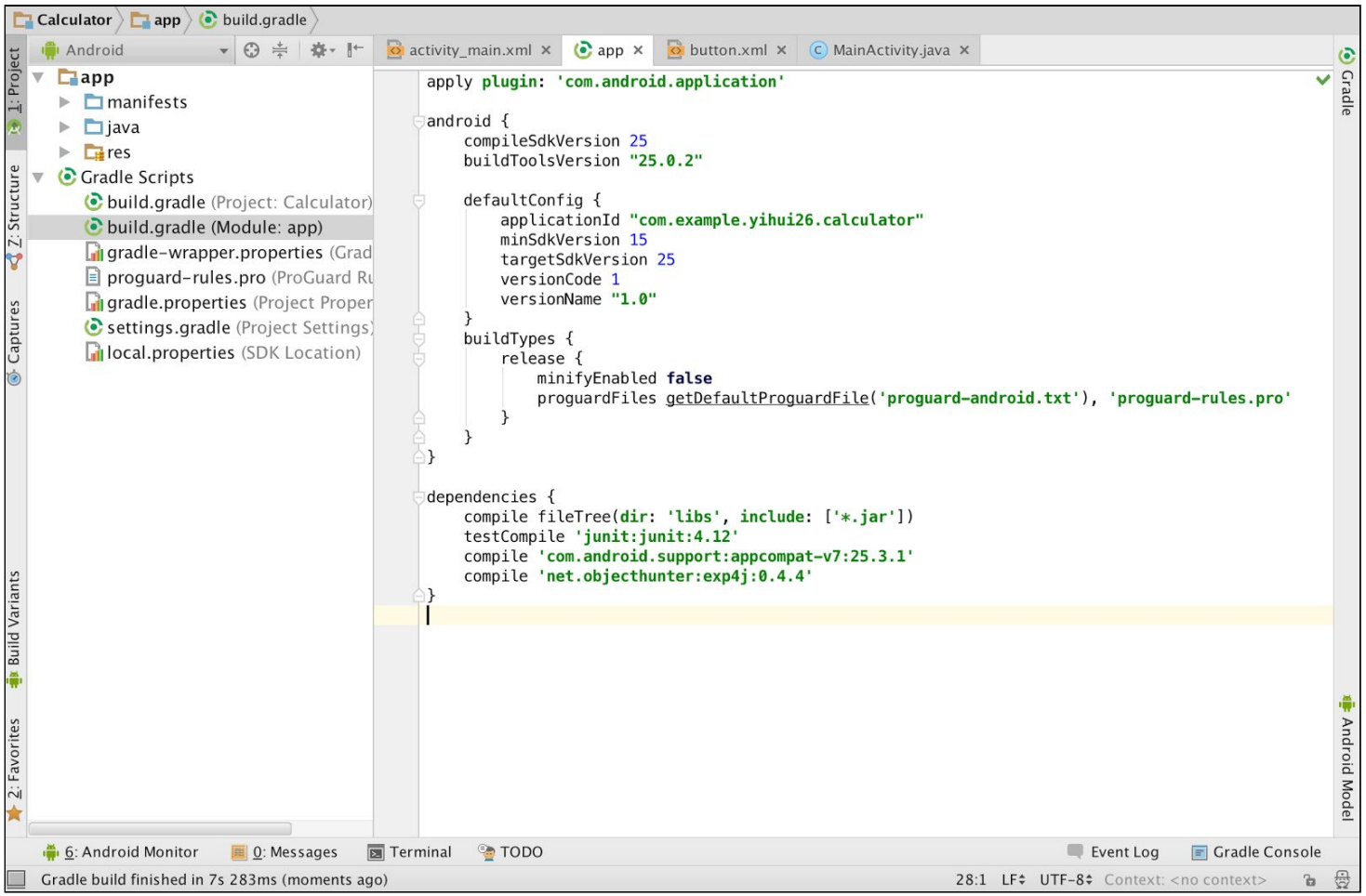


Step 11: Add exp4J library at “build.gradle (Module:app)” to evaluate arithmetic expressions, and sync the Gradle files.

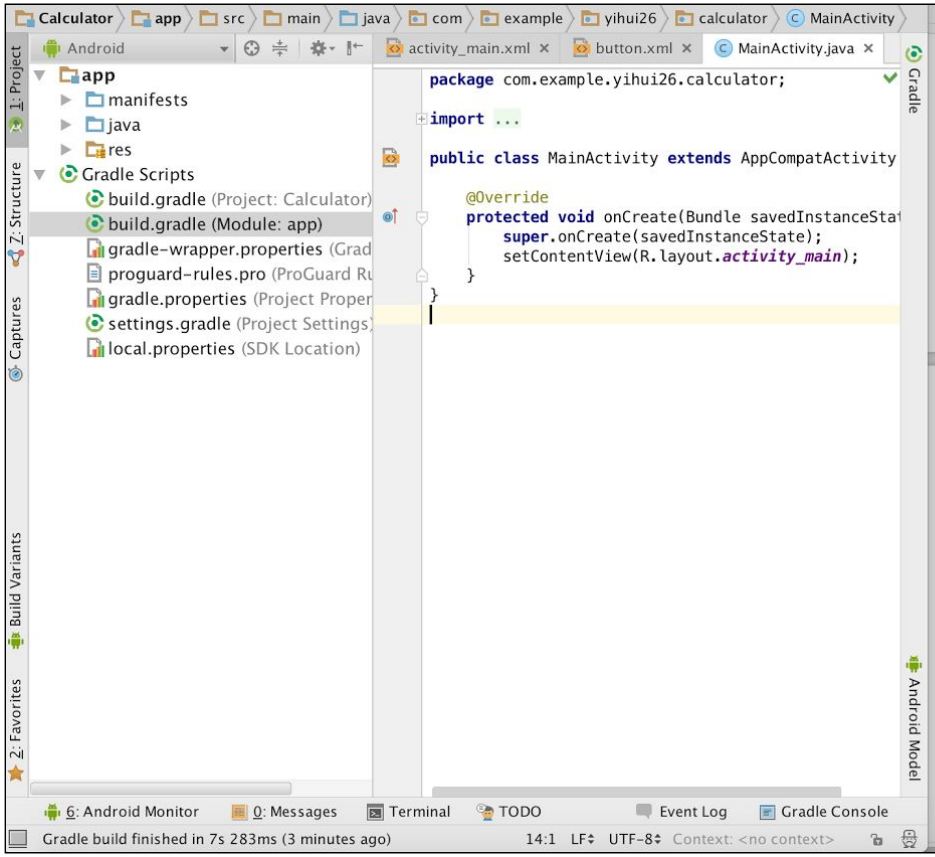

app - Calculator - [~/AndroidStudioProjects/Calculator]
Calculator
app
build.gradle
Android
activity_main.xml
app
button.xml
MainActivity.java
app
manifests
java
com.example.yihui26.calcula
MainActivity
com.example.yihui26.calcula
com.example.yihui26.calcula
res
drawable
layout
mipmap
values
Gradle Scripts
build.gradle (Project: Calculator)
build.gradle (Module: app)
gradle-wrapper.properties (Grad
proguard-rules.pro (ProGuard Ru
gradle.properties (Project Proper
settings.gradle (Project Settings)
local.properties (SDK Location)
apply plugin: 'com.android.application'
android {
compileSdkVersion 25
buildToolsVersion "25.0.2"
defaultConfig {
applicationId "com.example.yihui26.calculator"
minSdkVersion 15
targetSdkVersion 25
versionCode 1
versionName "1.0"
}
buildTypes {
release {
minifyEnabled false
proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
}
}
}
dependencies {
compile fileTree(dir: 'libs', include: ['*.jar'])
testCompile 'junit:junit:4.12'
compile 'com.android.support:appcompat-v7:25.3.1'
}
1: Project
2: Favorites
Build Variants
Captures
Structure
Gradle
Android Model
6: Android Monitor
0: Messages
Terminal
TODO
Event Log
Gradle Console
Gradle build finished in 18s 405ms (today 7:48 PM)
1:1
LF
UTF-8
Context: <no context>

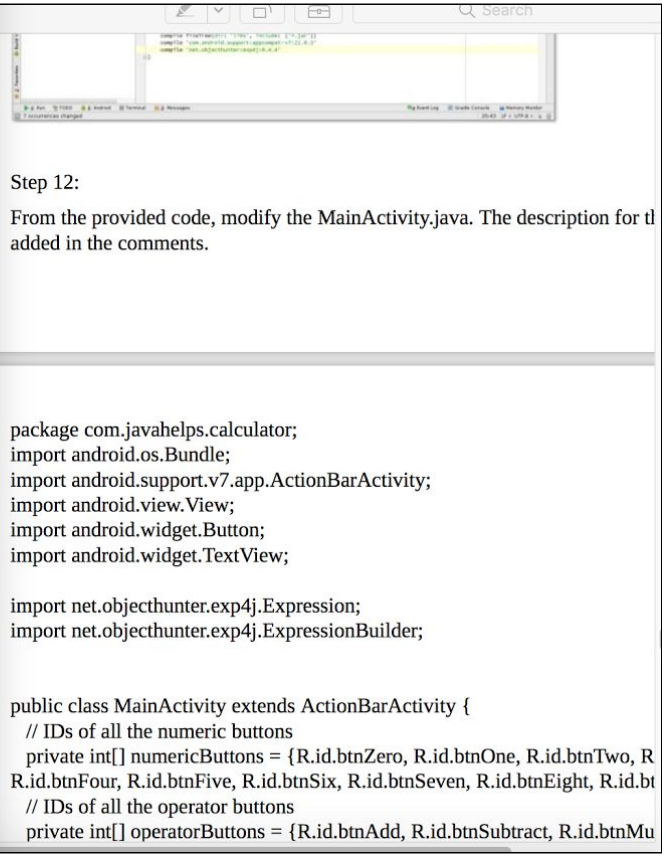

app - Calculator - [~/AndroidStudioProjects/Calculator]
Calculator
app
build.gradle
Android
activity_main.xml
app
button.xml
MainActivity.java
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to w... Sync Now
apply plugin: 'com.android.application'
android {
compileSdkVersion 25
buildToolsVersion "25.0.2"
defaultConfig {
applicationId "com.example.yihui26.calculator"
minSdkVersion 15
targetSdkVersion 25
versionCode 1
versionName "1.0"
}
buildTypes {
release {
minifyEnabled false
proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
}
}
}
dependencies {
compile fileTree(dir: 'libs', include: ['*.jar'])
testCompile 'junit:junit:4.12'
compile 'com.android.support:appcompat-v7:25.3.1'
compile 'net.objecthunter:exp4j:0.4.4'
}
app
manifests
java
com.example.yihui26.calcula
MainActivity
com.example.yihui26.calcula
com.example.yihui26.calcula
res
drawable
layout
mipmap
values
Gradle Scripts
build.gradle (Project: Calculator)
build.gradle (Module: app)
gradle-wrapper.properties (Grad
proguard-rules.pro (ProGuard Ru
gradle.properties (Project Proper
settings.gradle (Project Settings)
local.properties (SDK Location)
1: Project
Z: Structure
Captures
Build Variants
2: Favorites
Gradle
Android Model
6: Android Monitor
0: Messages
Terminal
TODO
Event Log
Gradle Console
Gradle build finished in 18s 405ms (today 7:48 PM)
38 chars
26:5
LF
UTF-8
Context: <no context>


activity_main.xml
app
button.xml
MainActivity.java
Gradle project sync in progress...
apply plugin: 'com.android.application'
Gradle

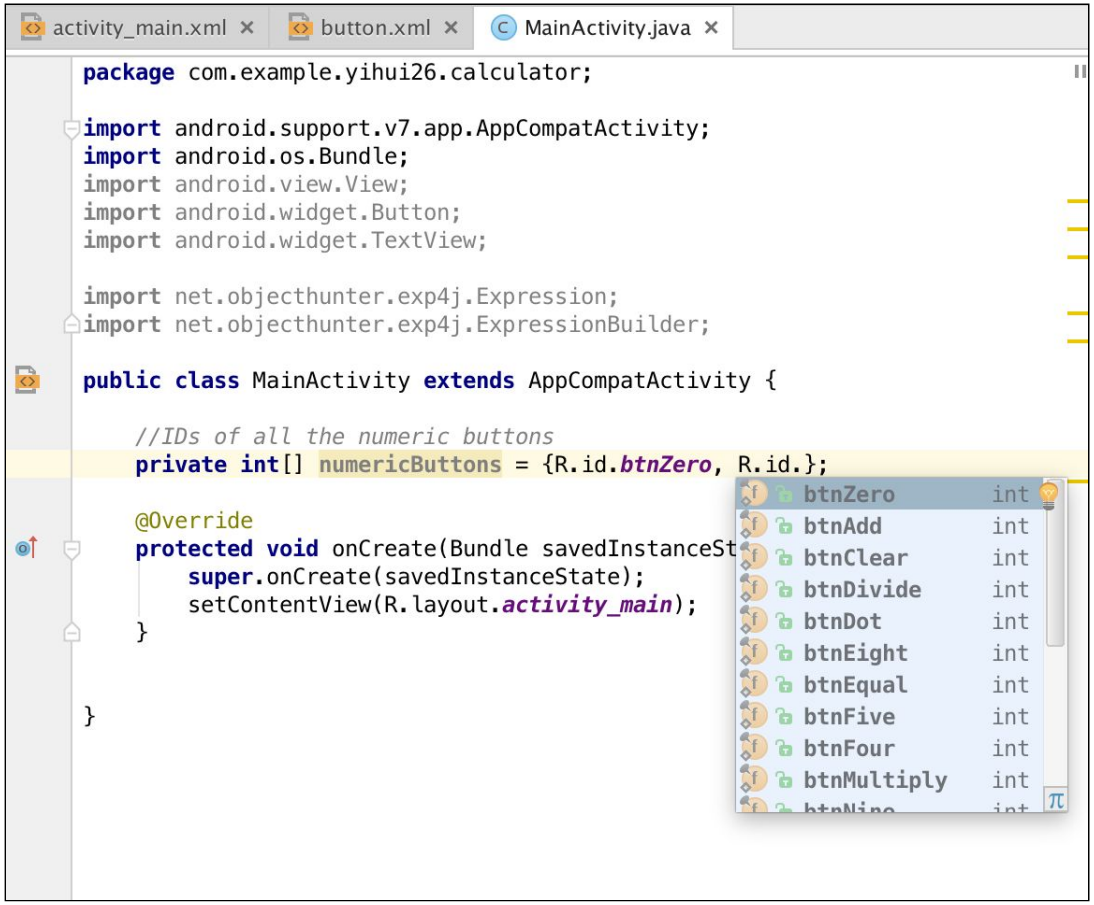

Calculator
app
build.gradle
Android
activity_main.xml
app
button.xml
MainActivity.java
app
manifests
java
res
Gradle Scripts
build.gradle (Project: Calculator)
build.gradle (Module: app)
gradle-wrapper.properties (Grad
proguard-rules.pro (ProGuard Ru
gradle.properties (Project Proper
settings.gradle (Project Settings)
local.properties (SDK Location)
apply plugin: 'com.android.application'
android {
compileSdkVersion 25
buildToolsVersion "25.0.2"
defaultConfig {
applicationId "com.example.yihui26.calculator"
minSdkVersion 15
targetSdkVersion 25
versionCode 1
versionName "1.0"
}
buildTypes {
release {
minifyEnabled false
proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
}
}
}
dependencies {
compile fileTree(dir: 'libs', include: ['*.jar'])
testCompile 'junit:junit:4.12'
compile 'com.android.support:appcompat-v7:25.3.1'
compile 'net.objecthunter:exp4j:0.4.4'
}
1: Project
7: Structure
Captures
Build Variants
2: Favorites
Gradle
Android Model
6: Android Monitor
0: Messages
Terminal
TODO
Event Log
Gradle Console
Gradle build finished in 7s 283ms (moments ago)
28:1
LF
UTF-8
Context: <no context>

Step 12: Update the logic at MainActivity.java to support calculator functionalities.

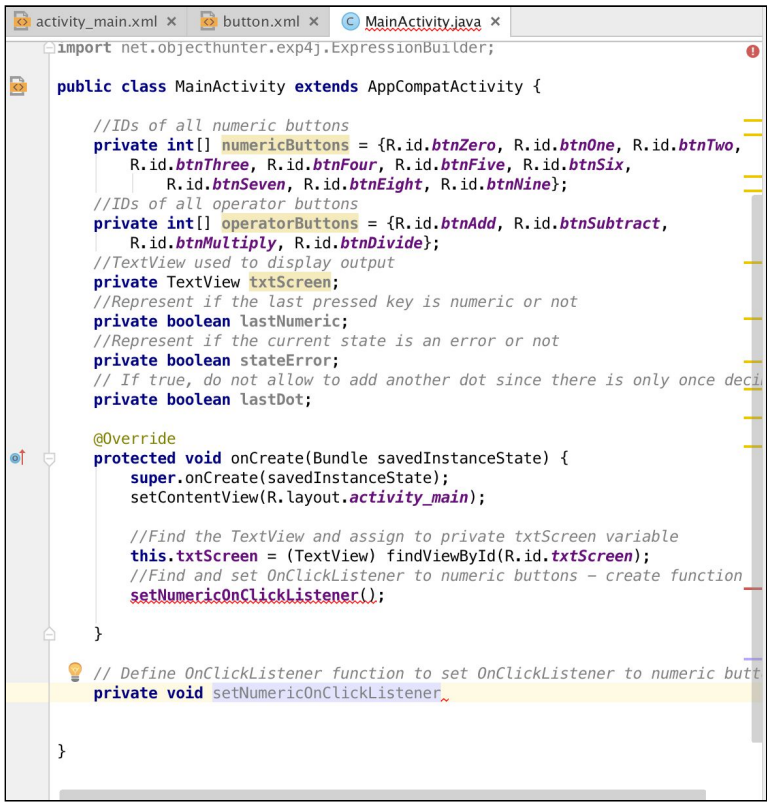




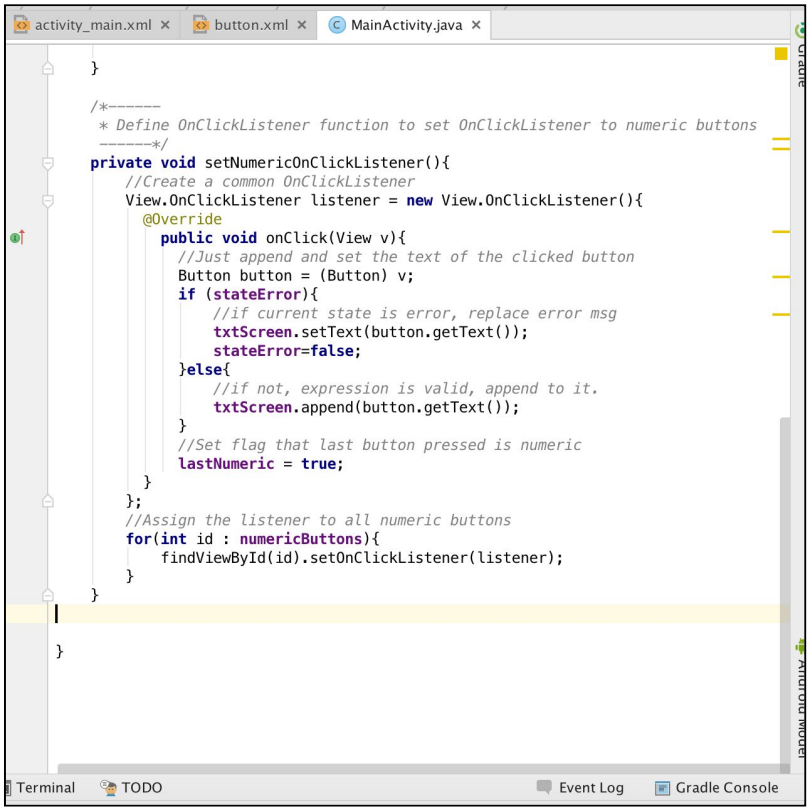
Define all the private variables.



Create a new function to set OnClickListener to numeric buttons.

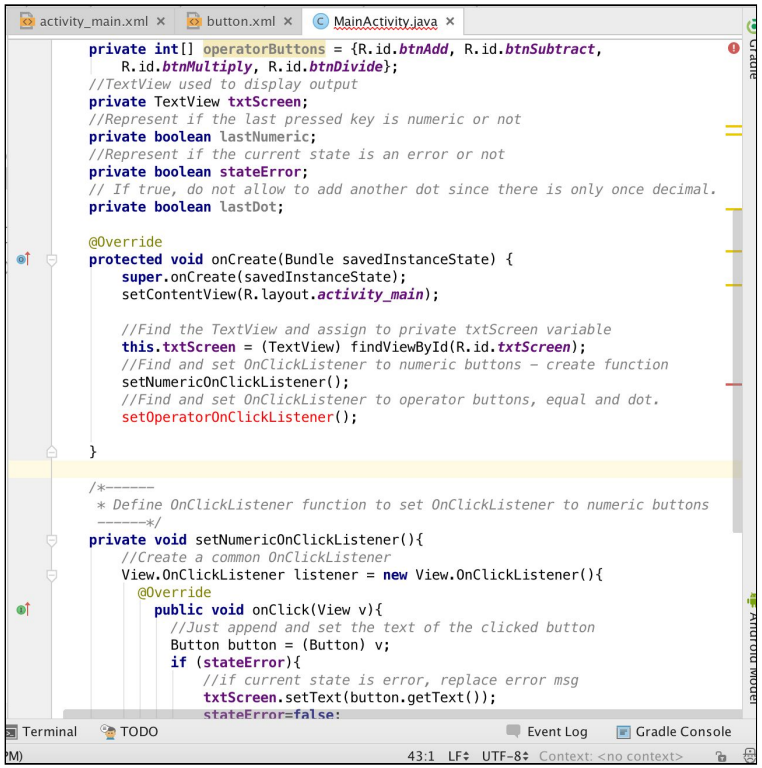


```
import net.objecthunter.exp4j.ExpressionBuilder;

public class MainActivity extends AppCompatActivity {

    //IDs of all numeric buttons
    private int[] numericButtons = {R.id.btnZero, R.id.btnOne, R.id.btnTwo,
        R.id.btnThree, R.id.btnFour, R.id.btnFive, R.id.btnSix,
            R.id.btnSeven, R.id.btnEight, R.id.btnNine};
    //IDs of all operator buttons
    private int[] operatorButtons = {R.id.btnAdd, R.id.btnSubtract,
        R.id.btnMultiply, R.id.btnDivide};
    //TextView used to display output
    private TextView txtScreen;
    //Represent if the last pressed key is numeric or not
    private boolean lastNumeric;
    //Represent if the current state is an error or not
    private boolean stateError;
    // If true, do not allow to add another dot since there is only once deci
    private boolean lastDot;

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

        //Find the TextView and assign to private txtScreen variable
        this.txtScreen = (TextView) findViewById(R.id.txtScreen);
        //Find and set OnClickListener to numeric buttons - create function
        setNumericOnClickListener();

    }

    // Define OnClickListener function to set OnClickListener to numeric butt
    private void setNumericOnClickListener

}
```

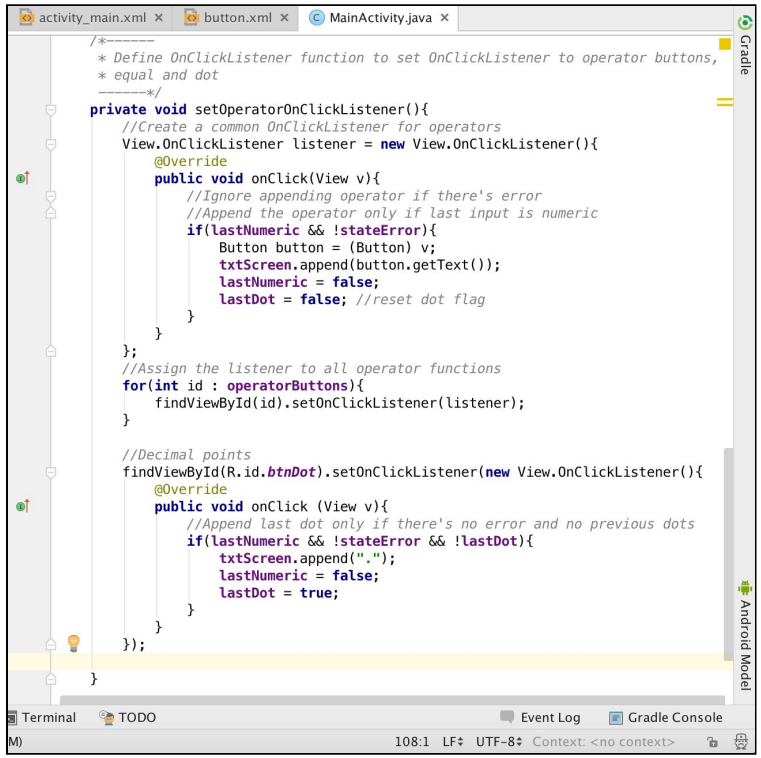


```
    }

    /*------
     * Define OnClickListener function to set OnClickListener to numeric buttons
     ------*/
    private void setNumericOnClickListener(){
        //Create a common OnClickListener
        View.OnClickListener listener = new View.OnClickListener(){
          @Override
            public void onClick(View v){
              //Just append and set the text of the clicked button
              Button button = (Button) v;
              if (stateError){
                  //if current state is error, replace error msg
                  txtScreen.setText(button.getText());
                  stateError=false;
              }else{
                  //if not, expression is valid, append to it.
                  txtScreen.append(button.getText());
              }
              //Set flag that last button pressed is numeric
              lastNumeric = true;
            }
        };
        //Assign the listener to all numeric buttons
        for(int id : numericButtons){
            findViewById(id).setOnClickListener(listener);
        }
    }

}
```

Also set OnClickListener to operator buttons.

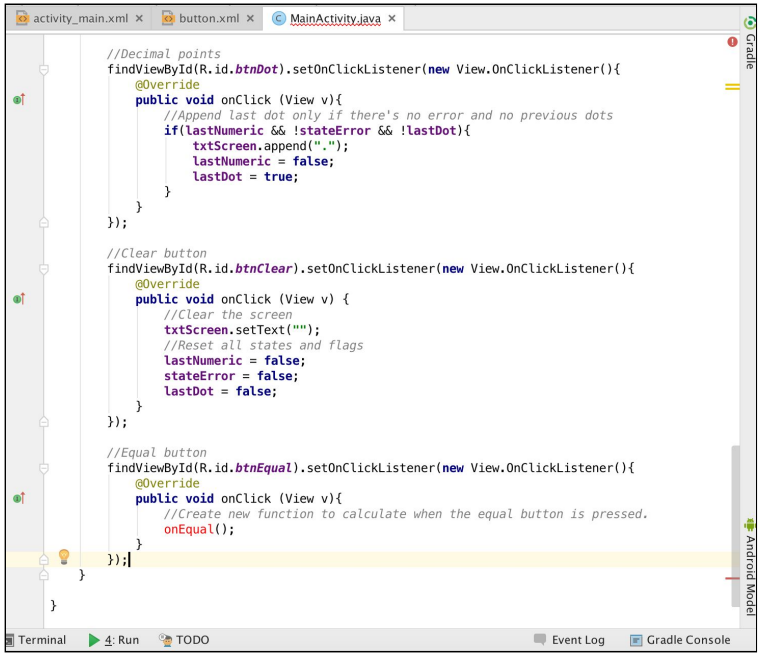


```
    private int[] operatorButtons = {R.id.btnAdd, R.id.btnSubtract,
        R.id.btnMultiply, R.id.btnDivide};
    //TextView used to display output
    private TextView txtScreen;
    //Represent if the last pressed key is numeric or not
    private boolean lastNumeric;
    //Represent if the current state is an error or not
    private boolean stateError;
    // If true, do not allow to add another dot since there is only once decimal.
    private boolean lastDot;

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

        //Find the TextView and assign to private txtScreen variable
        this.txtScreen = (TextView) findViewById(R.id.txtScreen);
        //Find and set OnClickListener to numeric buttons - create function
        setNumericOnClickListener();
        //Find and set OnClickListener to operator buttons, equal and dot.
        setOperatorOnClickListener();

    }

    /*------
     * Define OnClickListener function to set OnClickListener to numeric buttons
     ------*/
    private void setNumericOnClickListener(){
        //Create a common OnClickListener
        View.OnClickListener listener = new View.OnClickListener(){
          @Override
            public void onClick(View v){
              //Just append and set the text of the clicked button
              Button button = (Button) v;
              if (stateError){
                  //if current state is error, replace error msg
                  txtScreen.setText(button.getText());
                  stateError=false;
```

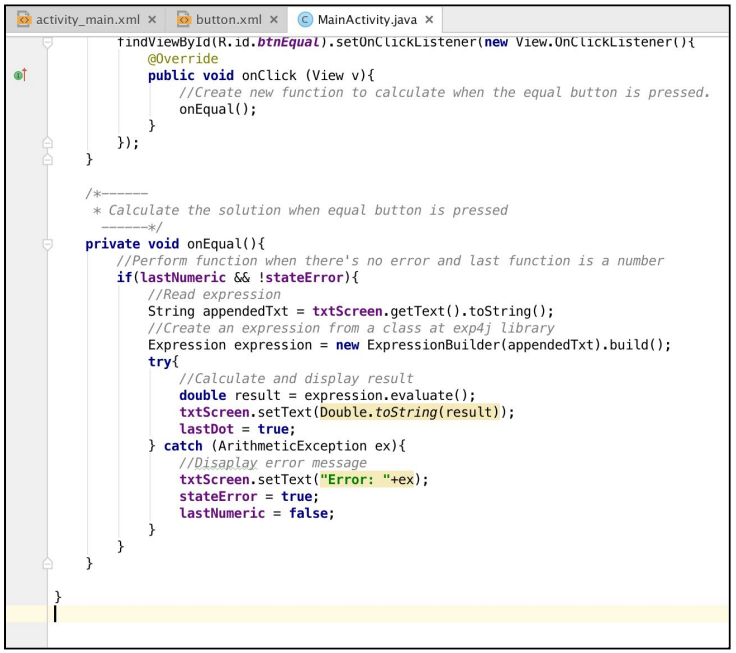


```
    /*------
     * Define OnClickListener function to set OnClickListener to operator buttons,
     * equal and dot
     ------*/
    private void setOperatorOnClickListener(){
        //Create a common OnClickListener for operators
        View.OnClickListener listener = new View.OnClickListener(){
            @Override
            public void onClick(View v){
                //Ignore appending operator if there's error
                //Append the operator only if last input is numeric
                if(lastNumeric && !stateError){
                    Button button = (Button) v;
                    txtScreen.append(button.getText());
                    lastNumeric = false;
                    lastDot = false; //reset dot flag
                }
            }
        };
        //Assign the listener to all operator functions
        for(int id : operatorButtons){
            findViewById(id).setOnClickListener(listener);
        }

        //Decimal points
        findViewById(R.id.btnDot).setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick (View v){
                //Append last dot only if there's no error and no previous dots
                if(lastNumeric && !stateError && !lastDot){
                    txtScreen.append(".");
                    lastNumeric = false;
                    lastDot = true;
                }
            }
        });

    }
```

Set up the equal function to perform calculation.

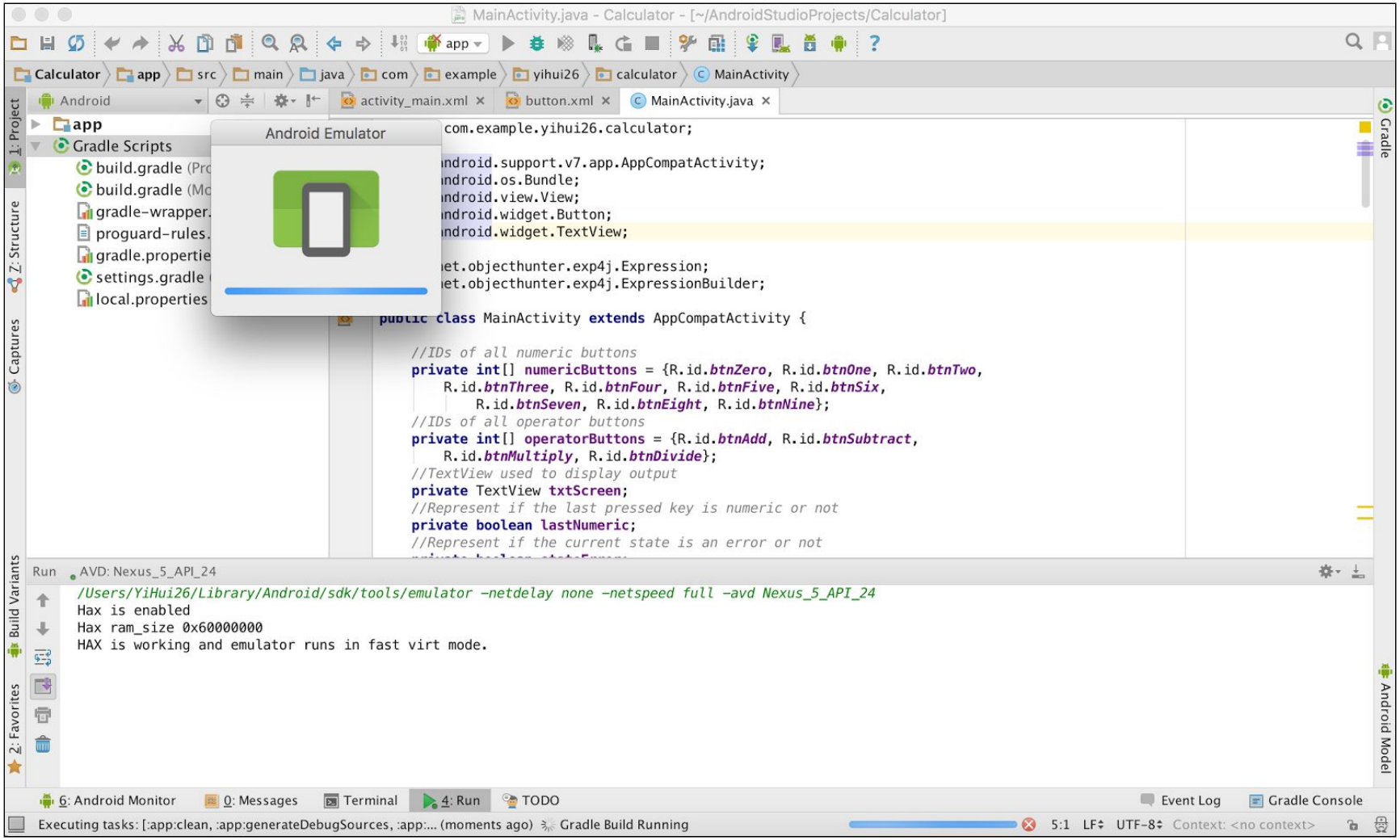


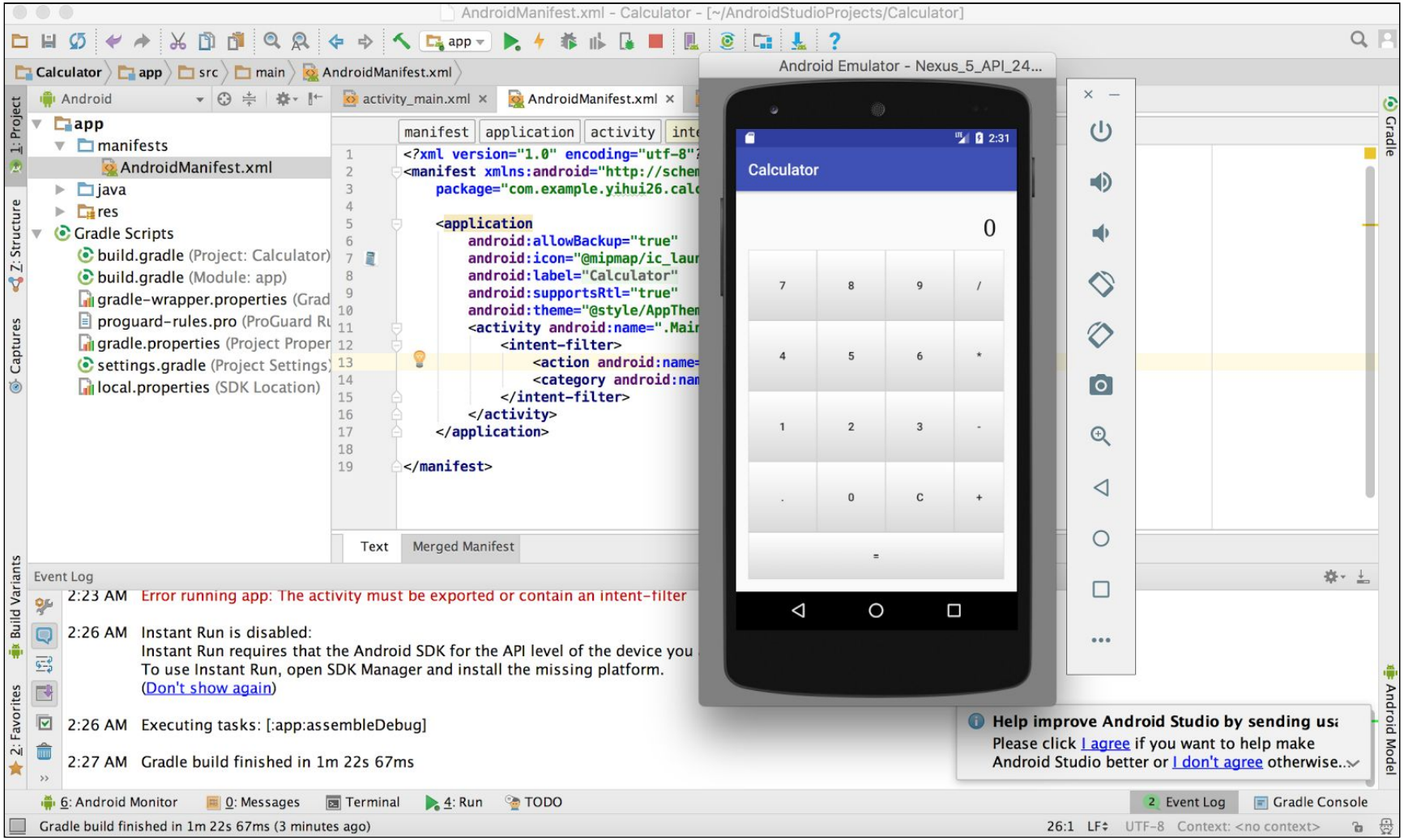
```
activity_main.xml × | button.xml × | MainActivity.java ×

        //Decimal points
        findViewById(R.id.btnDot).setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick (View v){
                //Append last dot only if there's no error and no previous dots
                if(lastNumeric && !stateError && !lastDot){
                    txtScreen.append(".");
                    lastNumeric = false;
                    lastDot = true;
                }
            }
        });

        //Clear button
        findViewById(R.id.btnClear).setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick (View v) {
                //Clear the screen
                txtScreen.setText("");
                //Reset all states and flags
                lastNumeric = false;
                stateError = false;
                lastDot = false;
            }
        });

        //Equal button
        findViewById(R.id.btnEqual).setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick (View v){
                //Create new function to calculate when the equal button is pressed.
                onEqual();
            }
        });
    }

}

Terminal | 4: Run | TODO | Event Log | Gradle Console
```



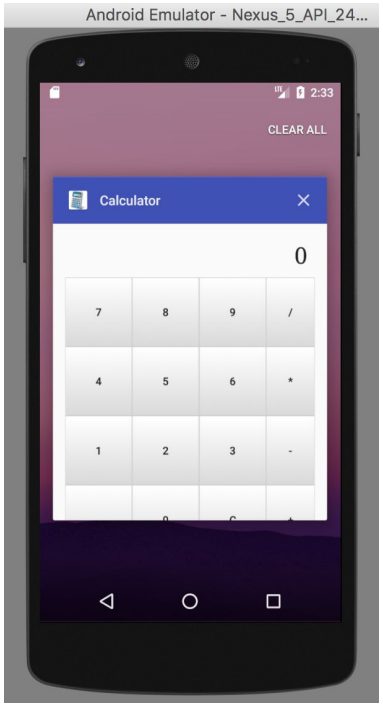
```
activity_main.xml × | button.xml × | MainActivity.java ×

        findViewById(R.id.btnEqual).setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick (View v){
                //Create new function to calculate when the equal button is pressed.
                onEqual();
            }
        });
    }

    /*------
     * Calculate the solution when equal button is pressed
      ------*/
    private void onEqual(){
        //Perform function when there's no error and last function is a number
        if(lastNumeric && !stateError){
            //Read expression
            String appendedTxt = txtScreen.getText().toString();
            //Create an expression from a class at exp4j library
            Expression expression = new ExpressionBuilder(appendedTxt).build();
            try{
                //Calculate and display result
                double result = expression.evaluate();
                txtScreen.setText(Double.toString(result));
                lastDot = true;
            } catch (ArithmeticException ex){
                //Disaplay error message
                txtScreen.setText("Error: "+ex);
                stateError = true;
                lastNumeric = false;
            }
        }
    }

}
```

Step 13: Run program in emulator




AndroidManifest.xml - Calculator - [~/AndroidStudioProjects/Calculator]
Android Emulator - Nexus_5_API_24...
Calculator
0
Project
Android
app
manifests
AndroidManifest.xml
java
res
Gradle Scripts
build.gradle (Project: Calculator)
build.gradle (Module: app)
gradle-wrapper.properties
proguard-rules.pro
gradle.properties
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
AndroidManifest.xml
Text
Merged Manifest
Event Log
2:23 AM Error running app: The activity must be exported or contain an intent-filter
2:26 AM Instant Run is disabled:
Instant Run requires that the Android SDK for the API level of the device you
To use Instant Run, open SDK Manager and install the missing platform.
(Don't show again)
2:26 AM Executing tasks: [:app:assembleDebug]
2:27 AM Gradle build finished in 1m 22s 67ms
Help improve Android Studio by sending us
Please click I agree if you want to help make
Android Studio better or I don't agree otherwise..
6: Android Monitor
0: Messages
Terminal
4: Run
TODO
Event Log
Gradle Console
Gradle build finished in 1m 22s 67ms (3 minutes ago)
26:1
LF
UTF-8
Context: <no context>

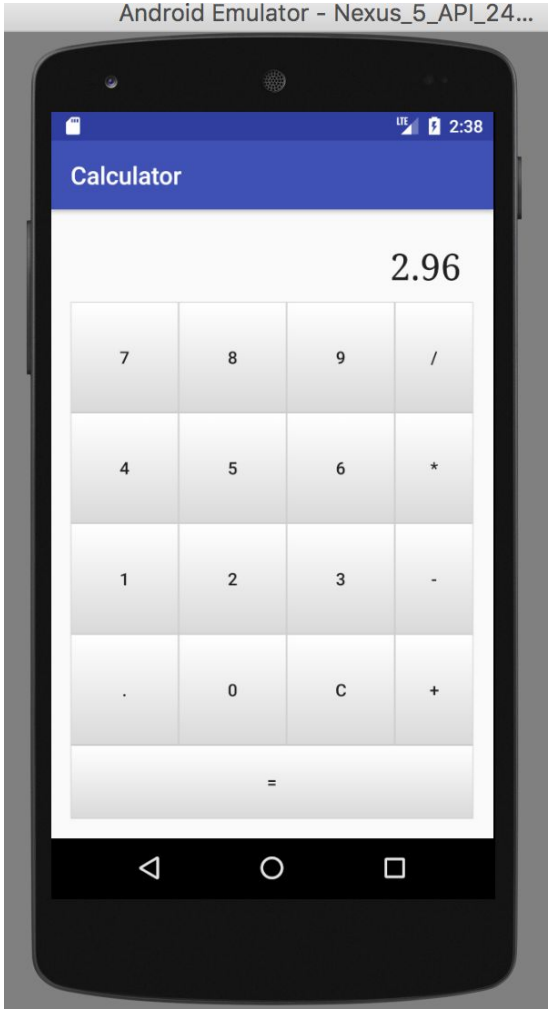
Calculator Unit Tests

1. App icon is nicely showing a calculator	2. Number entered and operators entered gets appended nicely. > Bug: Initial 0 gets appended, I want it to be replaced by the number I enter. > Fix: Numbers at number pad is small, I want to enlarge it.	3. Summation is correct.
		

4. Next parameter gets appended to answer.
Bug: New values should clear the screen.

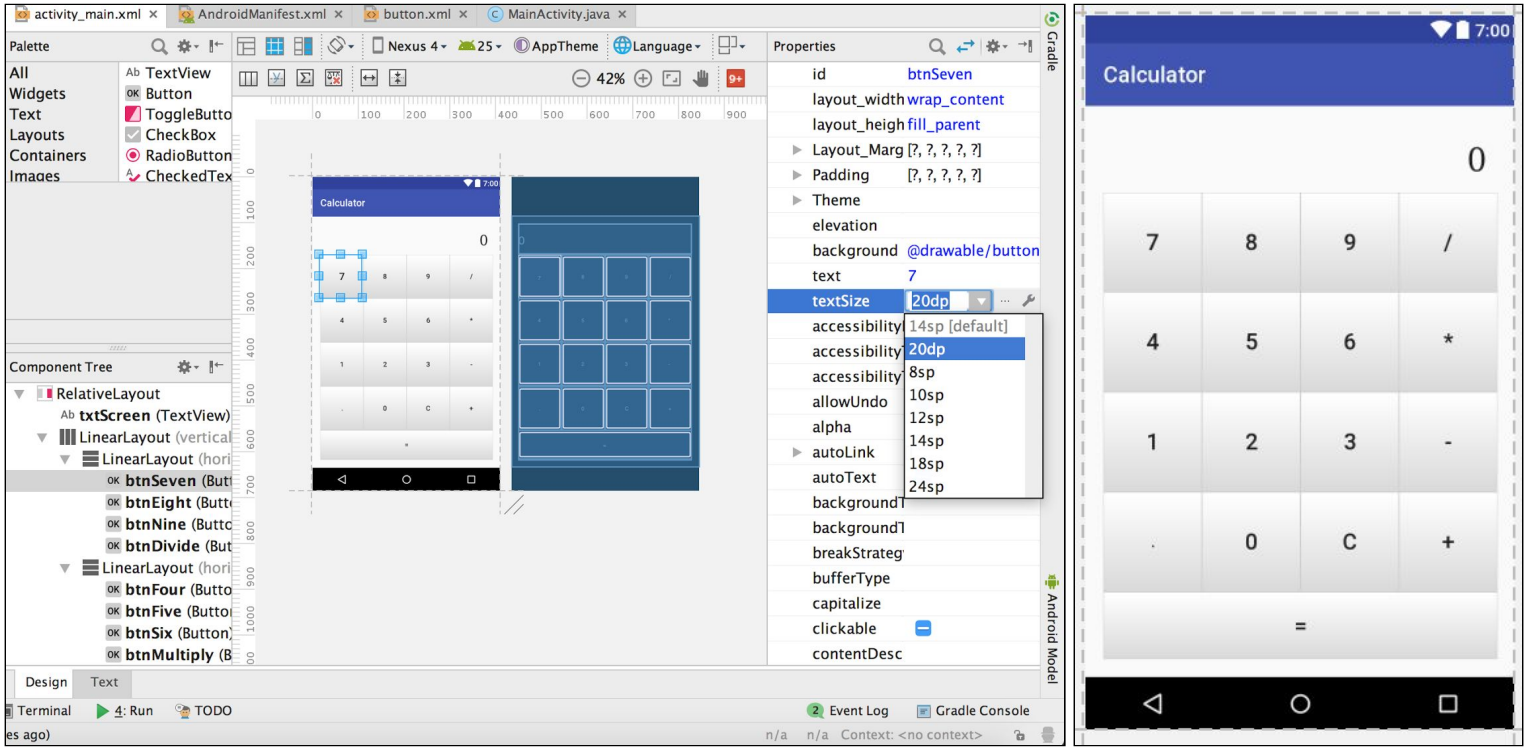
5. Additional new values are appended nicely.

6. Answer is correct.



Fixes

1. Text Size - Fixed by changing textSize value.
   Note: SDK changed because I upgraded to the latest 2.3.1 version for the Emulator to work



2. New values should clear the screen.
Fixed by introducing **private boolean firstNumber = true**;

1. Initial Screen	2. 0 gets cleared	3. Upon pressing equals, answer shows
		

4. if a number is pressed, it clears existing number.	5. Subsequent values append as usual.	6. Multiplication works correctly.
		

7. Upon pressing equals, if an operator is pressed, it appends. 	8. Division also works as expected 	9. Clear function works. Could default the value to 0 when clear, but shall leave it for now. 

Refined Codes for MainActivity.java

```
package com.example.yihui26.calculator;

import android.icu.text.DecimalFormat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import net.objecthunter.exp4j.Expression;
import net.objecthunter.exp4j.ExpressionBuilder;

public class MainActivity extends AppCompatActivity {

  //IDs of all numeric buttons
  private int[] numericButtons = {R.id.btnZero, R.id.btnOne, R.id.btnTwo,
      R.id.btnThree, R.id.btnFour, R.id.btnFive, R.id.btnSix,
          R.id.btnSeven, R.id.btnEight, R.id.btnNine};
  //IDs of all operator buttons
  private int[] operatorButtons = {R.id.btnAdd, R.id.btnSubtract,
      R.id.btnMultiply, R.id.btnDivide};
  //TextView used to display output
  private TextView txtScreen;
  //Represent if the last pressed key is numeric or not
  private boolean lastNumeric;
  //Represent if the current state is an error or not
  private boolean stateError;
  // If true, do not allow to add another dot since there is only once decimal.
  private boolean lastDot;
  // Indicate if it is the first value
  private boolean firstNumber = true;

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

```
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Find the TextView and assign to private txtScreen variable
    this.txtScreen = (TextView) findViewById(R.id.txtScreen);
    //Find and set OnClickListener to numeric buttons - create function
    setNumericOnClickListener();
    //Find and set OnClickListener to operator buttons, equal and dot.
    setOperatorOnClickListener();

}

/*------
 * Define OnClickListener function to set OnClickListener to numeric buttons
 ------*/
private void setNumericOnClickListener(){
    //Create a common OnClickListener
    View.OnClickListener listener = new View.OnClickListener(){
      @Override
        public void onClick(View v){
          //Just append and set the text of the clicked button
          Button button = (Button) v;
          if (stateError){
              //if current state is error, replace error msg
              txtScreen.setText(button.getText());
              stateError=false;
          }else{

              //check if should clear screen
              if (firstNumber){
                  txtScreen.setText("");
                  firstNumber = false;
              }
              //if not, expression is valid, append to it.
              txtScreen.append(button.getText());
```

```
            }
            //Set flag that last button pressed is numeric
            lastNumeric = true;
        }
    };
    //Assign the listener to all numeric buttons
    for(int id : numericButtons){
        findViewById(id).setOnClickListener(listener);
    }
}

/*------
 * Define OnClickListener function to set OnClickListener to operator buttons,
 * equal and dot
 ------*/
private void setOperatorOnClickListener(){
    //Create a common OnClickListener for operators
    View.OnClickListener listener = new View.OnClickListener(){
        @Override
        public void onClick(View v){
            //Ignore appending operator if there's error
            //Append the operator only if last input is numeric
            if(lastNumeric && !stateError){
                Button button = (Button) v;
                txtScreen.append(button.getText());
                lastNumeric = false;
                lastDot = false; //reset dot flag
            }
        }
    };
    //Assign the listener to all operator functions
    for(int id : operatorButtons){
        findViewById(id).setOnClickListener(listener);
    }

    //Decimal points
```

```
    findViewById(R.id.btnDot).setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick (View v){
            //Append last dot only if there's no error and no previous dots
            if(lastNumeric && !stateError && !lastDot){
                txtScreen.append(".");
                lastNumeric = false;
                lastDot = true;
            }
        }
    });

    //Clear button
    findViewById(R.id.btnClear).setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick (View v) {
            //Clear the screen
            txtScreen.setText("");
            //Reset all states and flags
            lastNumeric = false;
            stateError = false;
            lastDot = false;
        }
    });

    //Equal button
    findViewById(R.id.btnEqual).setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick (View v){
            //Create new function to calculate when the equal button is pressed.
            onEqual();
        }
    });
}

/*------
```

```
     * Calculate the solution when equal button is pressed
      ------*/
    private void onEqual(){
        //Perform function when there's no error and last function is a number
        if(lastNumeric && !stateError){
            //Read expression
            String appendedTxt = txtScreen.getText().toString();
            //Create an expression from a class at exp4j library
            Expression expression = new ExpressionBuilder(appendedTxt).build();
            try{
                double result = expression.evaluate();
                txtScreen.setText(Double.toString(result));
                lastDot = false;
                firstNumber = true;
            } catch (ArithmeticException ex){
                //Disaplay error message
                txtScreen.setText("Error: "+ex);
                stateError = true;
                lastNumeric = false;
            }
        }
    }

}
```