

Slide 7.3: Downloading and installing Android Studio

Slide 7.5: Android overview

Home



Android Programming

Before programming for Android, you need [Downloading and Installing Android Studio](#). The following list gives the steps of how to program for Android by using Android Studio. Developers may program the following files:

- `AndroidManifest.xml`, which includes the essential information about your app to the Android system, information the system must have before it can run any of the app's code,
- `MainActivity.java`, which is the main activity in Java,
- `NextActivity.java`, which is the activity of the next page in Java,
- `activity_main.xml`, which is the layout page of the `MainActivity.java`,
- `activity_next.xml`, which is the layout page of the `NextActivity.java`,
- `menu_main.xml`, which is the menu page of the `MainActivity.java`,
- `menu_next.xml`, which is the menu page of the `NextActivity.java`, and
- `strings.xml`, which is the file containing string values.

Basically, the two examples

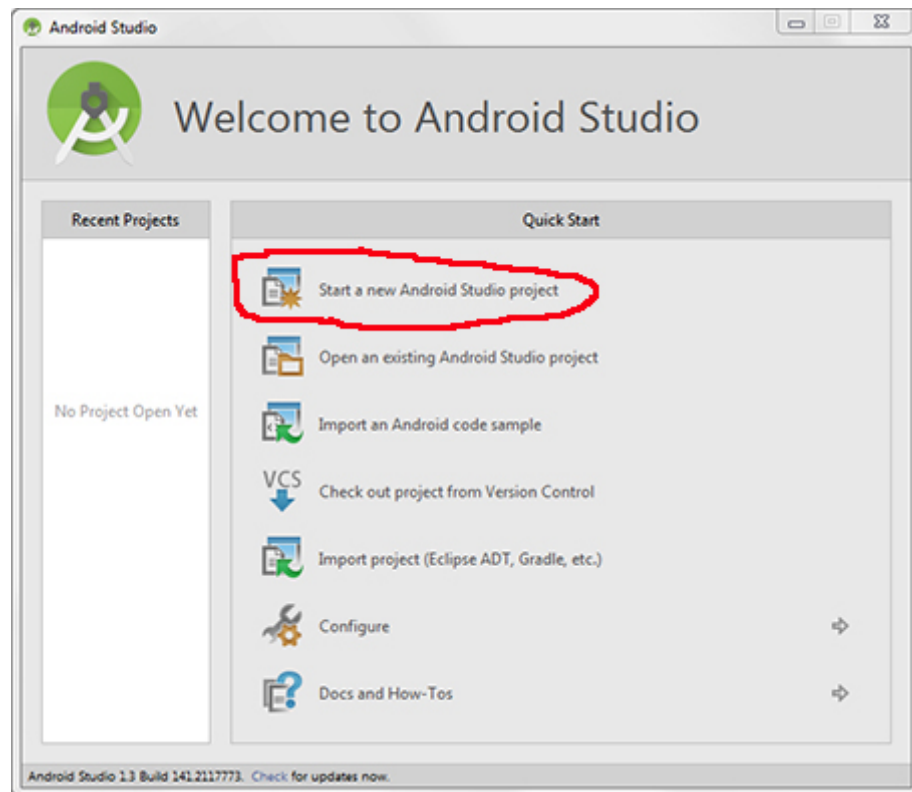
- [Android Adventures–Getting Started with Android Studio 1.0](#) and
- [Google Android Lessons](#)

are used in this page:

1. Launch the Android Studio.

Launch the Android Studio by selecting the following Windows options:

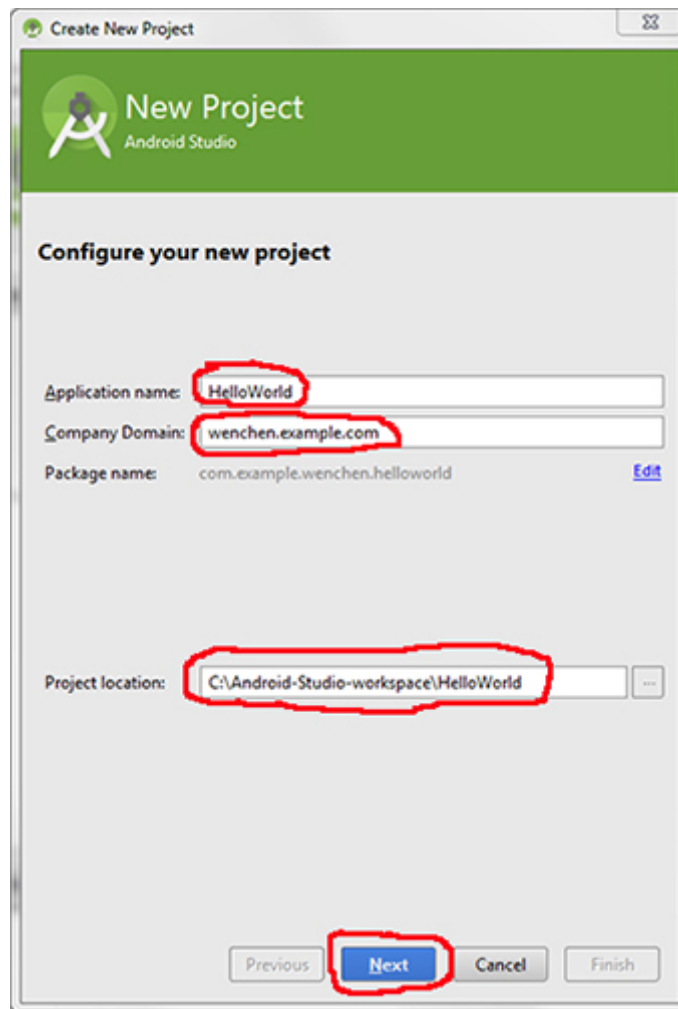
Start ⇒ All Programs ⇒ Android Studio ⇒ Android Studio



2. Start a New Android Studio Project.

Take the following steps:

- a. Fill in options for the new project such as
 - Application name: HelloWorld,
 - Company Domain: wenchen.example.com,
 - Package name: com.example.wenchen.helloworld (given), and
 - Project location: C:\Android-Studio-workspace\HelloWorld.



- b. Select the form factors the app will run on such as
- Phone and Tablet and
 - Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)
Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, the app will run on approximately 94.0% of the devices that are active on the Google Play Store.

Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately 94.0% of the devices that are active on the Google Play Store.

[Help me choose](#)

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

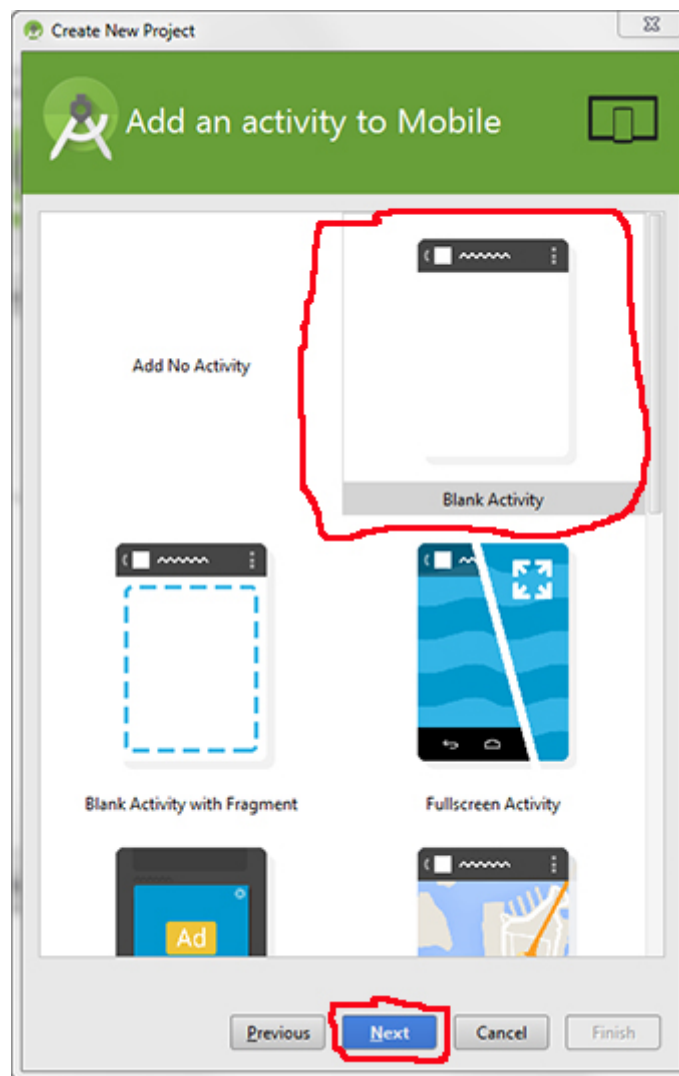
☐ Android Auto

☐ Glass (Not Installed) [Download](#)

Minimum SDK:

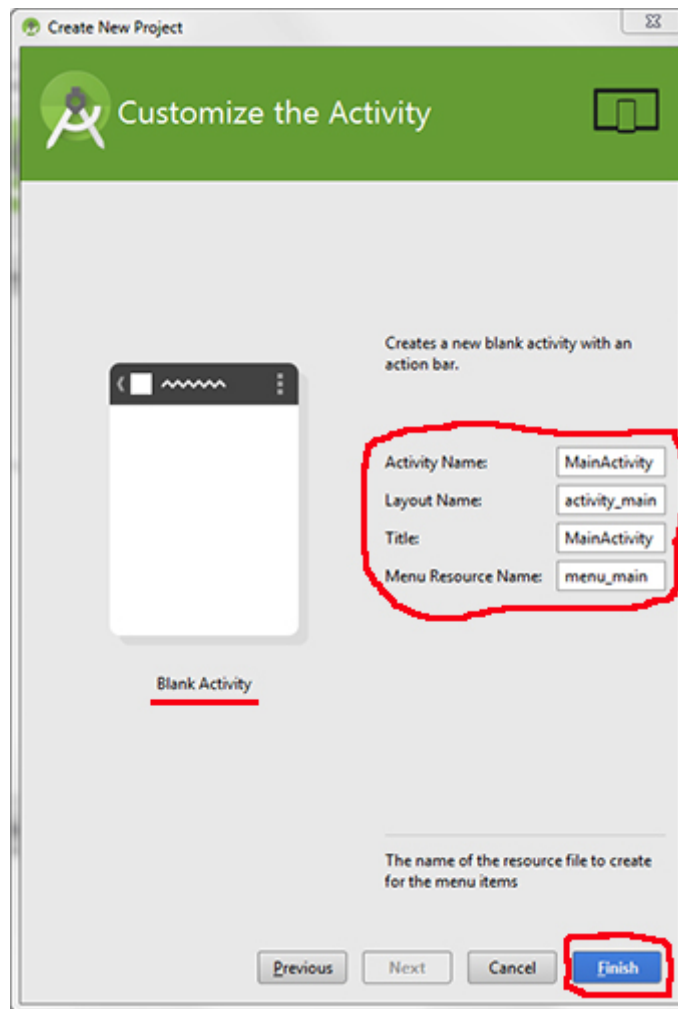
[Previous](#) **Next** [Cancel](#) [Finish](#)

c. Select a Blank Activity.



d. Fill in options such as

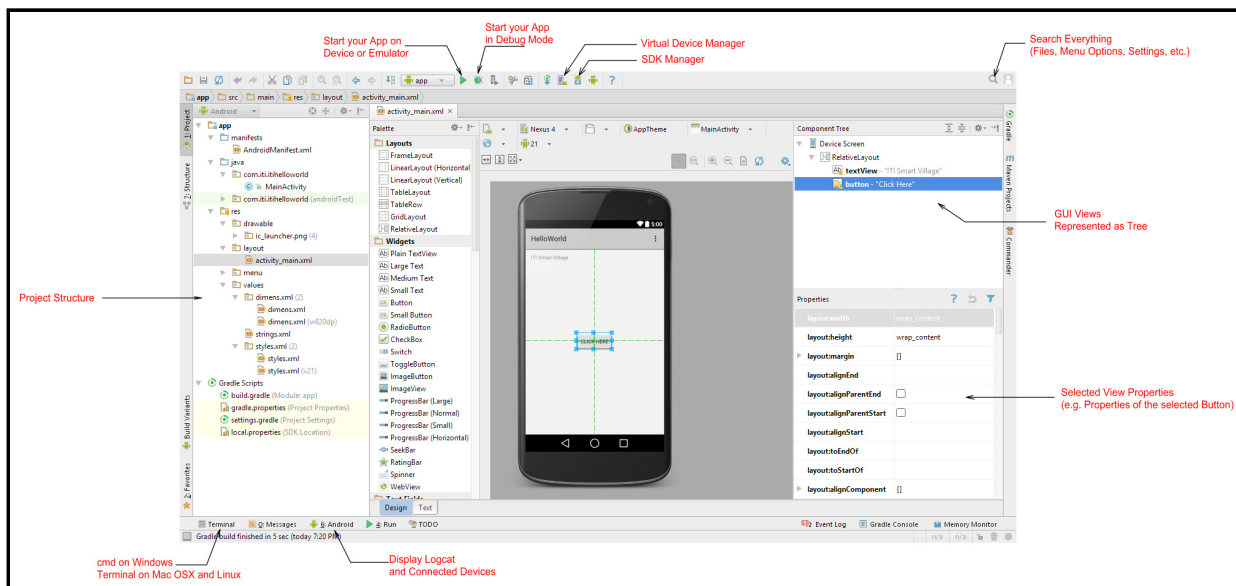
- Activity Name: MainActivity,
- Layout Name: activity_main,
- Title: MainActivity, and
- Menu Resource Name: menu_main.



3. Start Programming the Android Studio Project.

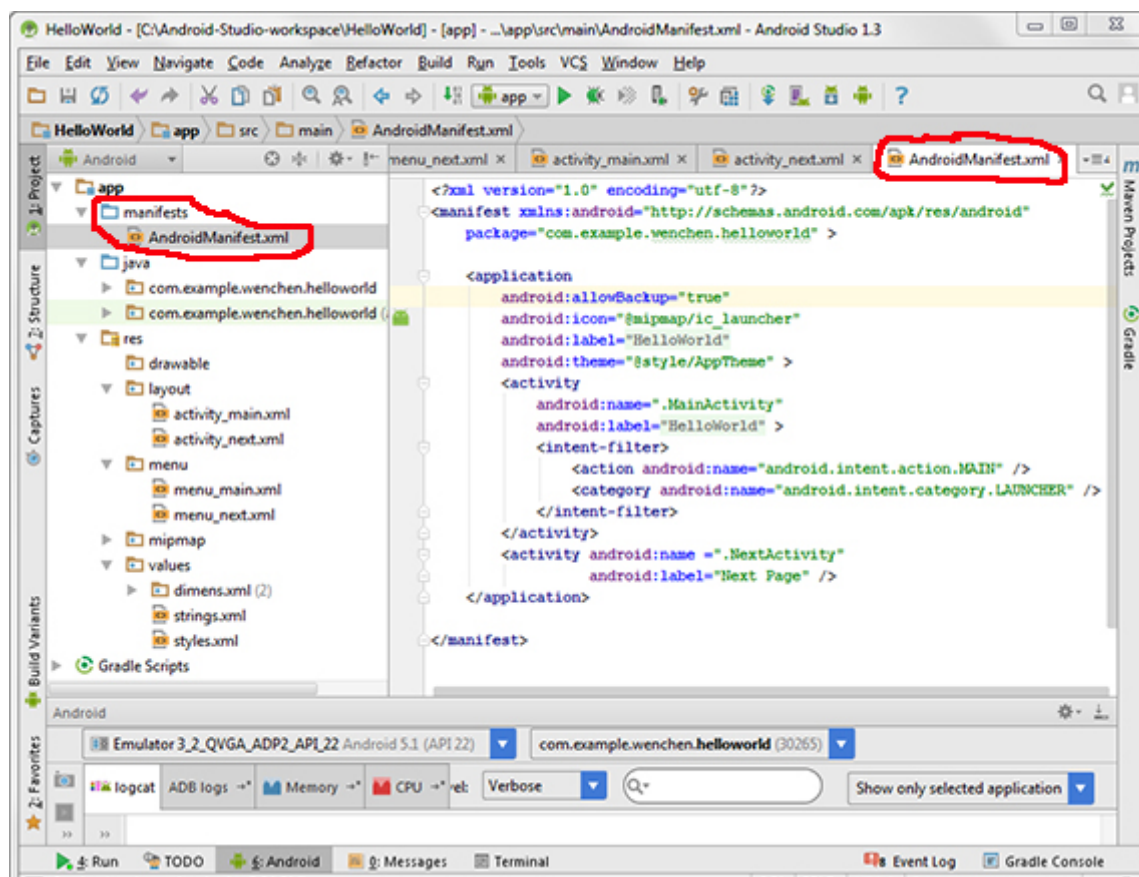
Note that Android is constantly revised and backward compatibility is an issue. This application works on API 15: Android 4.0.3 (IceCreamSandwich) correctly. If a newer version is used and minor compatibility errors show up, you should be able to fix them based on the error messages. One nice resource for Android debugging is [Stack Overflow](http://stackoverflow.com).

The Android Studio IDE (Integrated Development Environment) is as follows:



4. Program AndroidManifest.xml.

- Expand the app of the project HelloWorld in the left pane.
- Double click the file AndroidManifest.xml and complete it such as:



C:\Android-Studio-workspace\HelloWorld\app\src\main\AndroidManifest.xml

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
03     package="com.example.wenchen.helloworld" >
04
05     <application

```

```

06     android:allowBackup="true"
07     android:icon="@mipmap/ic_launcher"
08     android:label="@string/app_name"
09     android:theme="@style/AppTheme" >
10     <activity
11         android:name=".MainActivity"
12         android:label="@string/app_name" >
13         <intent-filter>
14             <action android:name="android.intent.action.MAIN" />
15             <category android:name="android.intent.category.LAUNCHER" />
16         </intent-filter>
17     </activity>
18     <activity android:name=".NextActivity"
19         android:label="Next Page" />
20 </application>
21 </manifest>

```

5. Program MainActivity.java.

Double click the file `java\com\example\wenchen\helloworld\MainActivity.java` and a template will be given as follows:

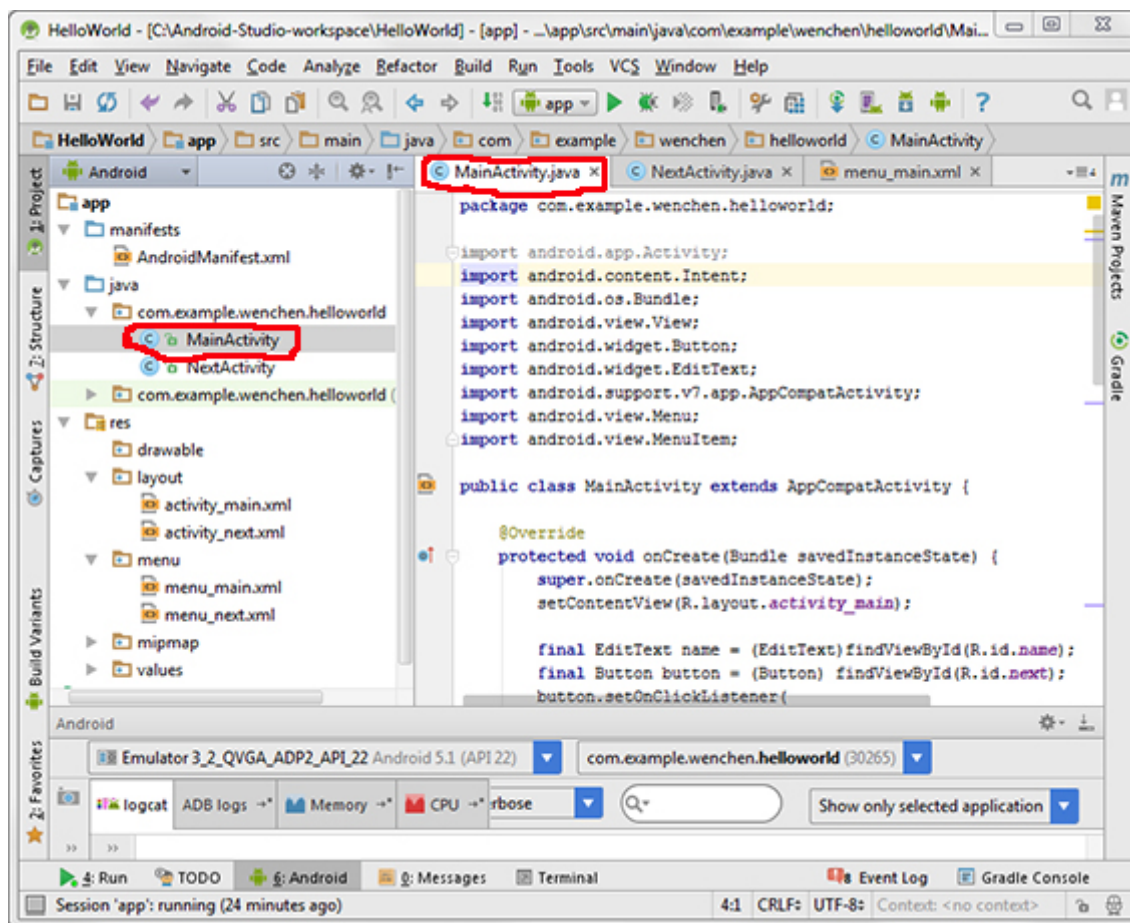
`C:\Android-Studio-workspace\HelloWorld\app\src\main\java\com\example\wenchen\helloworld\MainActivity.java`

```

01 package com.example.wenchen.helloworld;
02
03 import android.support.v7.app.AppCompatActivity;
04 import android.os.Bundle;
05 import android.view.Menu;
06 import android.view.MenuItem;
07
08 public class MainActivity extends AppCompatActivity {
09
10     @Override
11     protected void onCreate( Bundle savedInstanceState) {
12         super.onCreate( savedInstanceState );
13         setContentView( R.layout.activity_main );
14     }
15
16     @Override
17     public boolean onCreateOptionsMenu( Menu menu ) {
18         // Inflate the menu; this adds items to the action bar if it is present.
19         getMenuInflater( ).inflate( R.menu.menu_main, menu );
20         return true;
21     }
22
23     @Override
24     public boolean onOptionsItemSelected( MenuItem item ) {
25         // Handle action bar item clicks here. The action bar will
26         // automatically handle clicks on the Home/Up button, so long
27         // as you specify a parent activity in AndroidManifest.xml.
28         int id = item.getItemId( );
29
30         // noinspection SimplifiableIfStatement
31         if ( id == R.id.action_settings ) {
32             return true;
33         }
34         return super.onOptionsItemSelected( item );
35     }
36 }

```

Complete the Java class `MainActivity.java` such as



C:\Android-Studio-workspace\HelloWorld\app\src\main\java\com\example\wenchen\helloworld\MainActivity.java

```

01 package com.example.wenchen.helloworld;
02
03 import android.app.Activity;
04 import android.content.Intent;
05 import android.os.Bundle;
06 import android.view.View;
07 import android.widget.Button;
08 import android.widget.EditText;
09 import android.support.v7.app.AppCompatActivity;
10 import android.view.Menu;
11 import android.view.MenuItem;
12
13 public class MainActivity extends AppCompatActivity {
14
15     @Override
16     protected void onCreate( Bundle savedInstanceState ) {
17         super.onCreate( savedInstanceState );
18         setContentView( R.layout.activity_main );
19
20         final EditText name = (EditText) findViewById( R.id.name );
21         final Button button = (Button) findViewById( R.id.next );
22         button.setOnClickListener(
23             new View.OnClickListener() {
24                 public void onClick( View v ) {
25                     /** Here i calls a new screen. */
26                     Intent i = new Intent( MainActivity.this, NextActivity.class );
27                     i.putExtra( "name", name.getText().toString() );
28                     startActivity( i );
29                 }
30             }
31         );
32     }

```

```

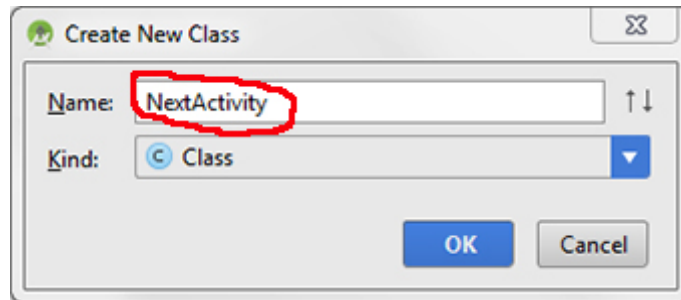
33
34 @Override
35 public boolean onCreateOptionsMenu( Menu menu ) {
36     // Inflate the menu; this adds items to the action bar if it is present.
37     getMenuInflater( ).inflate( R.menu.menu_main, menu );
38     return true;
39 }
40
41 @Override
42 public boolean onOptionsItemSelected( MenuItem item ) {
43     // Handle action bar item clicks here. The action bar will
44     // automatically handle clicks on the Home/Up button, so long
45     // as you specify a parent activity in AndroidManifest.xml.
46     int id = item.getItemId( );
47
48     //noinspection SimplifiableIfStatement
49     if ( id == R.id.action_settings ) {
50         return true;
51     }
52     else if ( id == R.id.next ) {
53         Intent i = new Intent( MainActivity.this, NextActivity.class );
54         startActivity( i );
55     }
56     return super.onOptionsItemSelected( item );
57 }
58 }

```

6. Program NextActivity.java.

Using the left pane, add another Java class, NextActivity.java, by right clicking the mouse:

com.example.wenchen.helloworld ⇒ New ⇒ Java Class



Complete the Java class NextActivity.java such as

C:\Android-Studio-workspace\HelloWorld\app\src\main\java\com\example\wenchen\helloworld\NextActivity.java

```

01 package com.example.wenchen.helloworld;
02
03 import android.app.Activity;
04 import android.content.Intent;
05 import android.os.Bundle;
06 import android.view.Menu;
07 import android.view.MenuItem;
08 import android.view.View;
09 import android.widget.Button;
10 import android.widget.TextView;
11
12 public class NextActivity extends Activity {
13     /** Called when the activity is first created. */
14     @Override
15     public void onCreate( Bundle savedInstanceState ) {
16         super.onCreate( savedInstanceState );

```

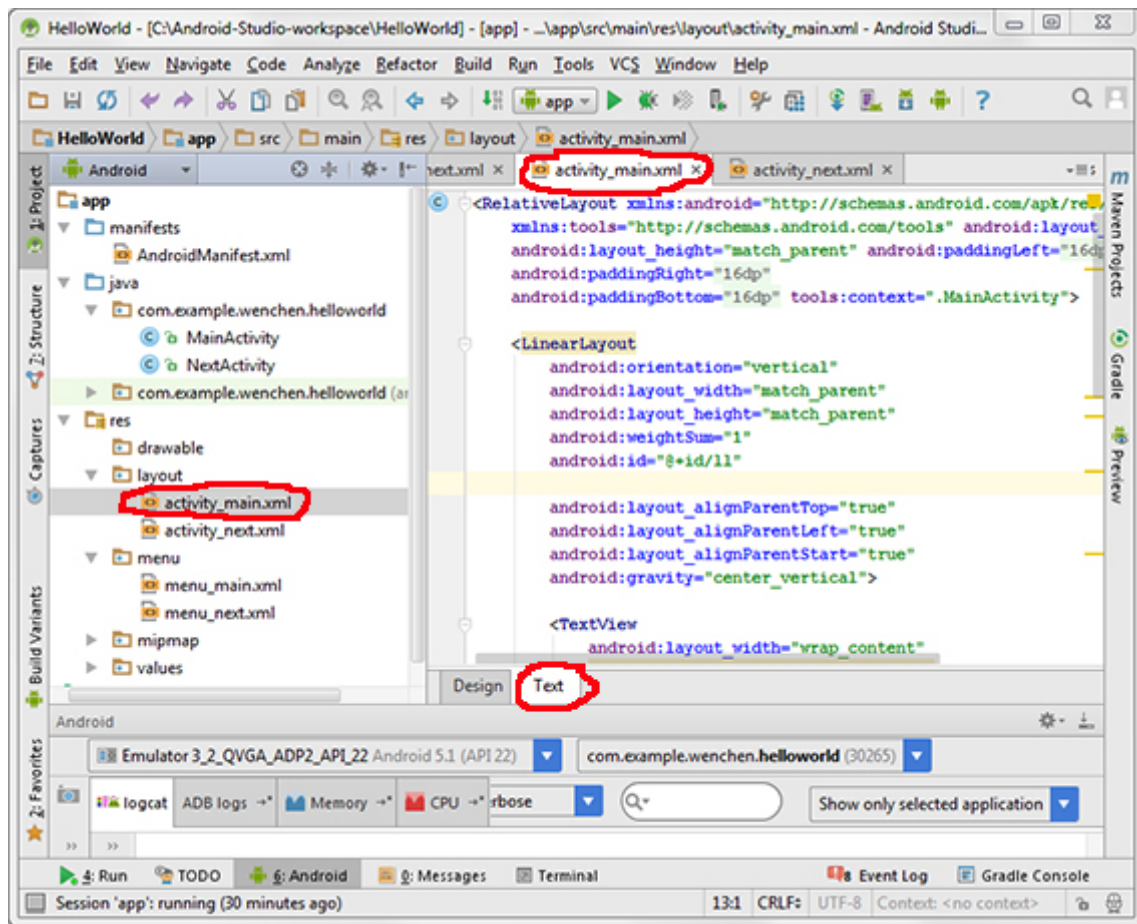
```

17 setContentView( R.layout.activity_next );
18
19 final TextView tvView = (TextView) findViewById( R.id.tvView );
20 Intent intent = getIntent( );
21 String name = intent.getStringExtra( "name" );
22 tvView.setText( "Welcome, " + name );
23 final Button button = (Button) findViewById( R.id.home );
24 button.setOnClickListener(
25     new View.OnClickListener( ) {
26         public void onClick( View v ) {
27             Intent i = new Intent( NextActivity.this, MainActivity.class );
28             startActivity( i );
29         }
30     }
31 );
32 }
33
34 @Override
35 public boolean onCreateOptionsMenu( Menu menu ) {
36     // Inflate the menu; this adds items to the action bar if it is present.
37     getMenuInflater( ).inflate( R.menu.menu_next, menu );
38     return true;
39 }
40
41 @Override
42 public boolean onOptionsItemSelected( MenuItem item ) {
43     // Handle action bar item clicks here. The action bar will
44     // automatically handle clicks on the Home/Up button, so long
45     // as you specify a parent activity in AndroidManifest.xml.
46     int id = item.getItemId( );
47
48     // noinspection SimplifiableIfStatement
49     if ( id == R.id.action_settings ) {
50         return true;
51     }
52     else if ( id == R.id.home ) {
53         Intent i = new Intent( NextActivity.this, MainActivity.class );
54         startActivity( i );
55     }
56     return super.onOptionsItemSelected( item );
57 }
58 }

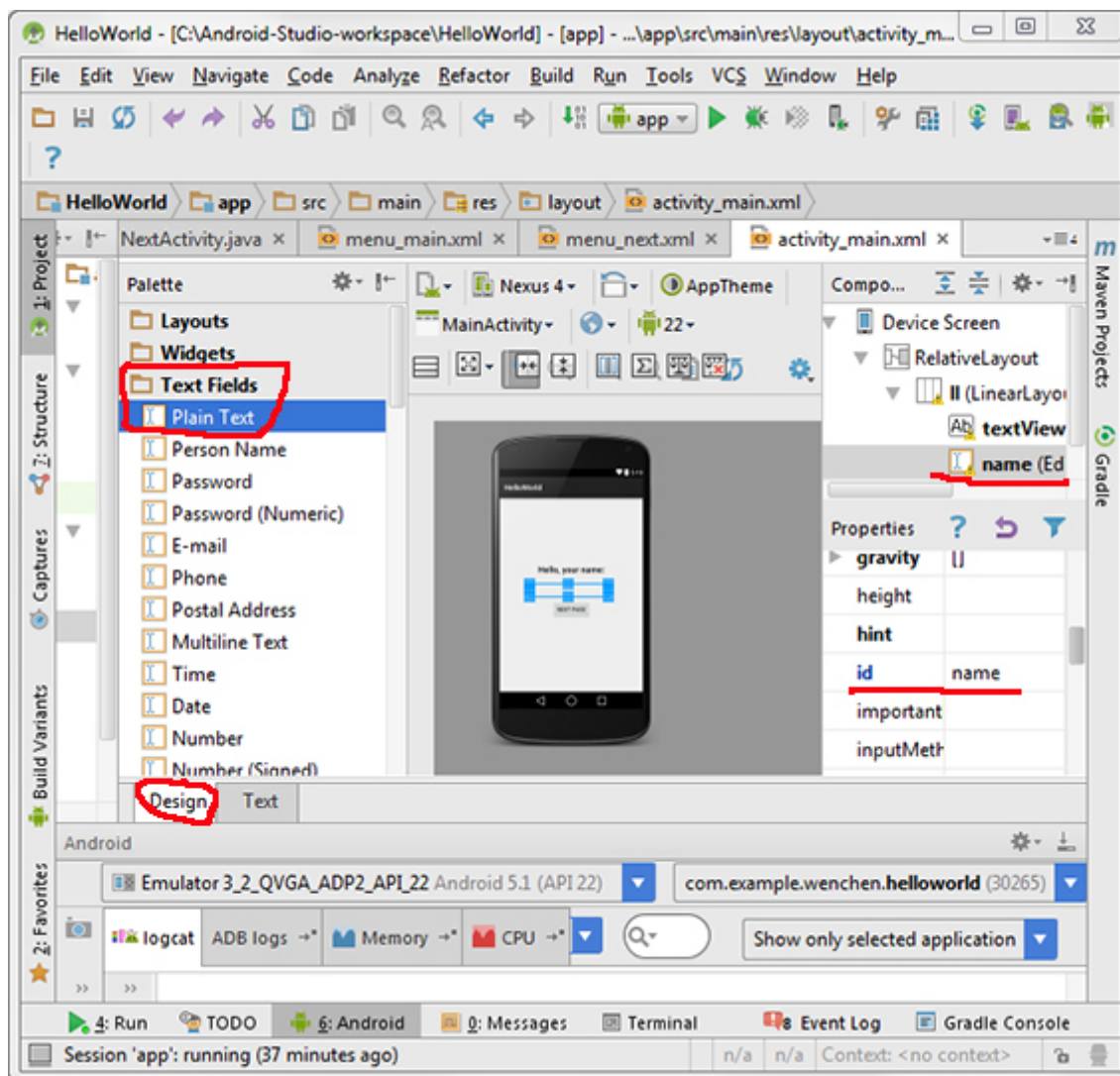
```

7. Program activity_main.xml.

Double click the file activity_main.xml and complete it such as:



Android Studio is a WYSIWYG (What You See Is What You Get) IDE. You may use the palette to draw the user interface and the Android Studio will convert it into XML:



C:\Android-Studio-workspace\HelloWorld\app\src\main\res\layout\activity_main.xml

```

01 <LinearLayout
02     xmlns:android = "http://schemas.android.com/apk/res/android"
03     android:orientation = "vertical"
04     android:layout_width = "match_parent"
05     android:layout_height = "match_parent"
06     android:weightSum = "1"
07     android:gravity = "center_vertical|center_horizontal">
08
09     <TextView
10         android:layout_width = "wrap_content"
11         android:layout_height = "wrap_content"
12         android:textAppearance = "?android:attr/textAppearanceLarge"
13         android:text = "Hello, your name:"
14         android:layout_weight = "0.07"
15         android:textStyle = "bold" />
16
17     <EditText
18         android:layout_width = "216dp"
19         android:layout_height = "wrap_content"
20         android:id = "@+id/name" />
21
22     <Button
23         android:layout_width = "wrap_content"
24         android:layout_height = "wrap_content"
25         android:text = "Next Page"
26         android:id = "@+id/next" />

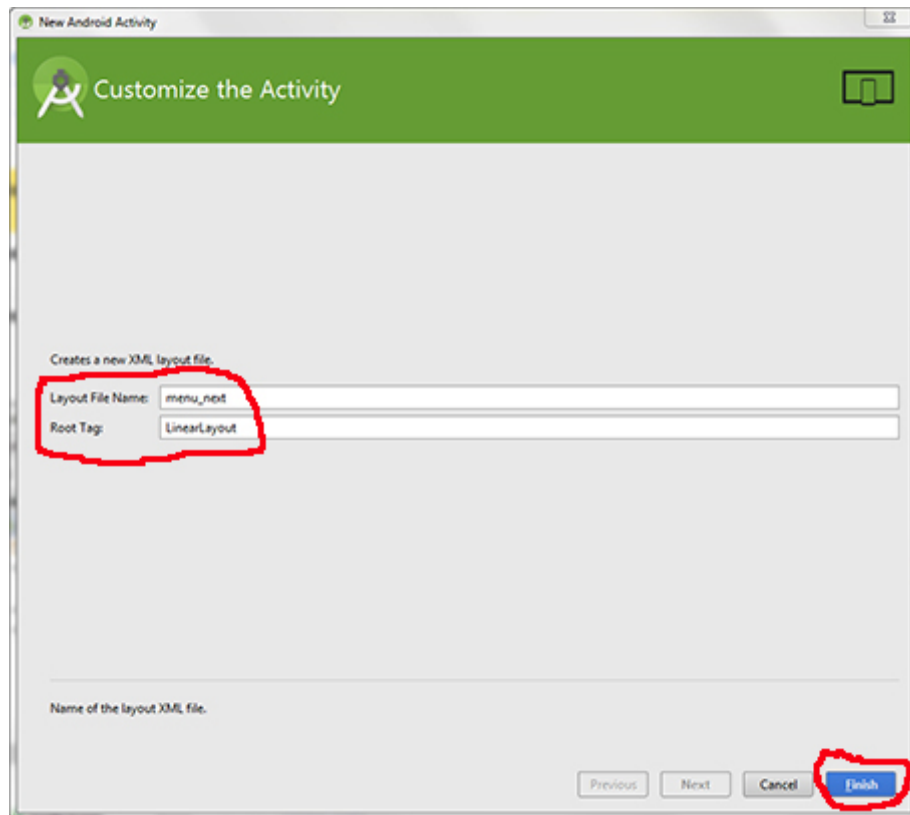
```

27
28 `</LinearLayout>`

8. Program activity_next.xml.

Using the left pane, add another layout XML file, activity_next.xml, by selecting the following options:

Right clicking the “layout ⇒ New ⇒ XML ⇒ layout XML file”



C:\Android-Studio-workspace\HelloWorld\app\src\main\res\activity_next.xml

```

01 <LinearLayout
02     xmlns:android            = "http://schemas.android.com/apk/res/android"
03     android:layout_width     = "match_parent"
04     android:layout_height    = "match_parent"
05     android:orientation      = "vertical"
06     android:weightSum        = "1"
07     android:gravity          = "center_vertical|center_horizontal">
08
09     <TextView
10         android:layout_width  = "287dp"
11         android:layout_height = "wrap_content"
12         android:textAppearance = "?android:attr/textAppearanceLarge"
13         android:id            = "@+id/tvView"
14         android:textStyle     = "bold"
15         android:layout_weight = "0.10"
16         android:gravity       = "center_horizontal" />
17
18     <Button
19         android:layout_width  = "wrap_content"
20         android:layout_height = "40dp"
21         android:text          = "Home Page"

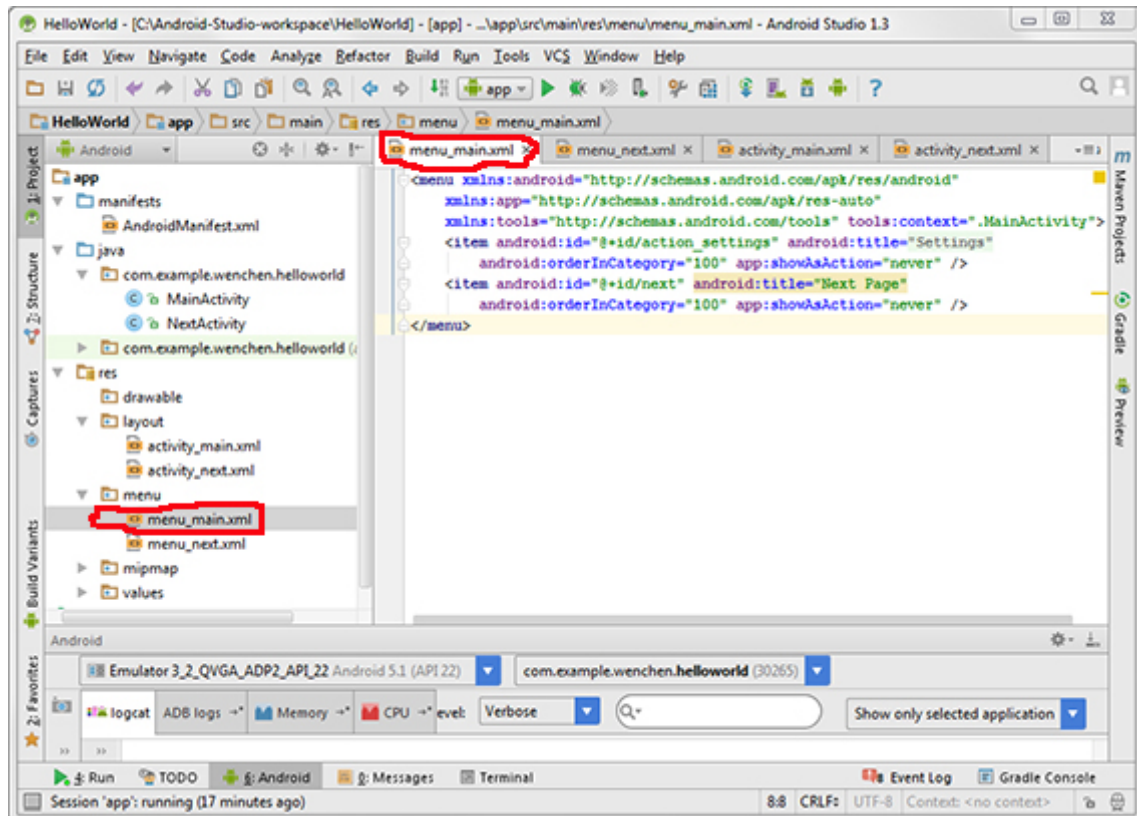
```



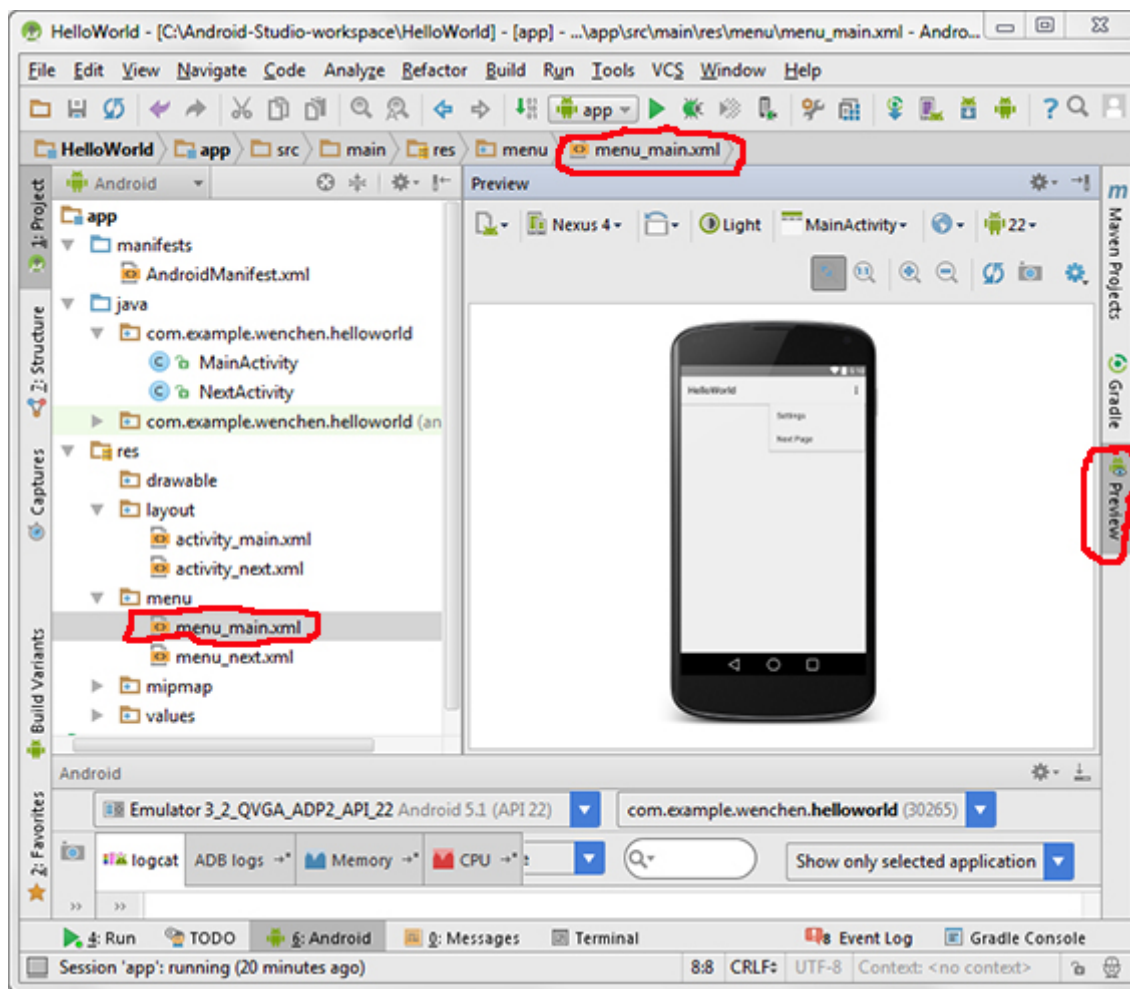
```
22     android:id               = "@+id/home" />  
23  
24 </LinearLayout>
```

9. Program menu_main.xml.

Double click the file menu_main.xml and complete it such as:



You may use the Android Studio to view the result:



C:\Android-Studio-workspace\HelloWorld\app\src\main\res\menu_main.xml

```

01 <menu
02     xmlns:android="http://schemas.android.com/apk/res/android"
03     xmlns:app="http://schemas.android.com/apk/res-auto"
04     xmlns:tools="http://schemas.android.com/tools"
05     tools:context=".MainActivity">
06
07     <item android:id="@+id/action_settings"
08         android:title="@string/action_settings"
09         android:orderInCategory="100"
10         app:showAsAction="never" />
11
12     <item android:id="@+id/next"
13         android:title="Next Page"
14         android:orderInCategory="100"
15         app:showAsAction="never" />
16 </menu>

```

10. Program menu_next.xml.

C:\Android-Studio-workspace\HelloWorld\app\src\main\res\menu_next.xml

```

01 <menu
02     xmlns:android="http://schemas.android.com/apk/res/android"
03     xmlns:app="http://schemas.android.com/apk/res-auto"
04     xmlns:tools="http://schemas.android.com/tools"

```



```

05 tools:context=".NextActivity">
06
07 <item android:id="@+id/action_settings"
08     android:title="@string/action_settings"
09     android:orderInCategory="100"
10     app:showAsAction="never" />
11
12 <item android:id="@+id/home"
13     android:title="Home Page"
14     android:orderInCategory="100"
15     app:showAsAction="never" />
16 </menu>

```

11. Program strings.xml.

The file defines some string variables such as

C:\Android-Studio-workspace\HelloWorld\app\src\main\res\values\strings.xml

```

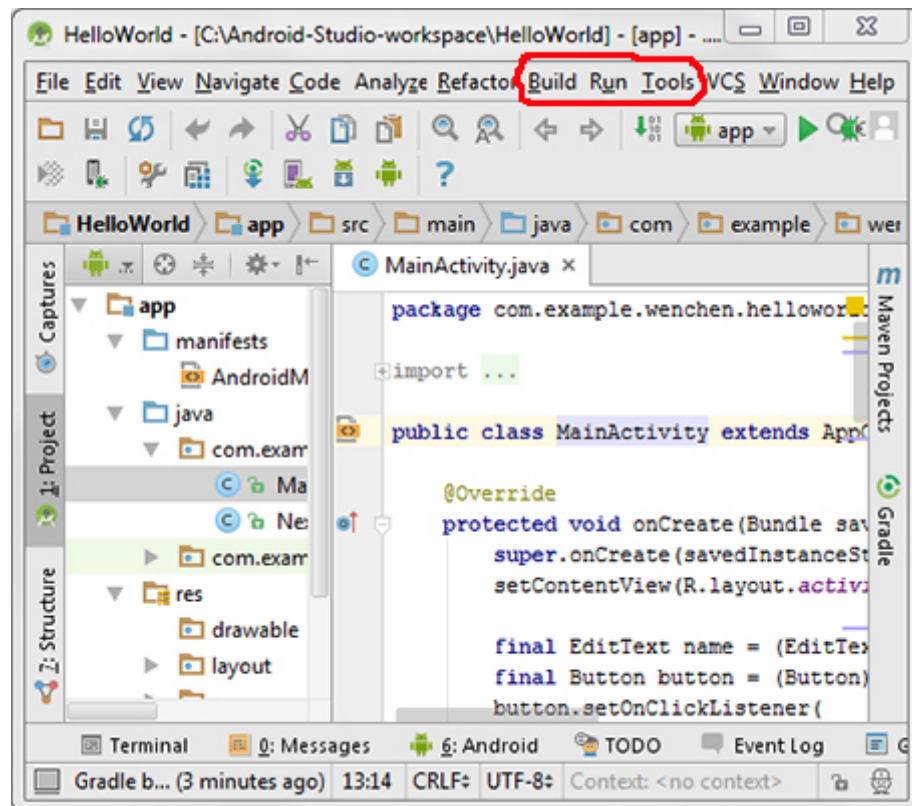
1 <resources>
2 <string name="app_name">HelloWorld</string>
3 <string name="hello_world">Hello world!</string>
4 <string name="action_settings">Settings</string>
5 </resources>

```

12. Build the App.

Build the project by selecting the following options:

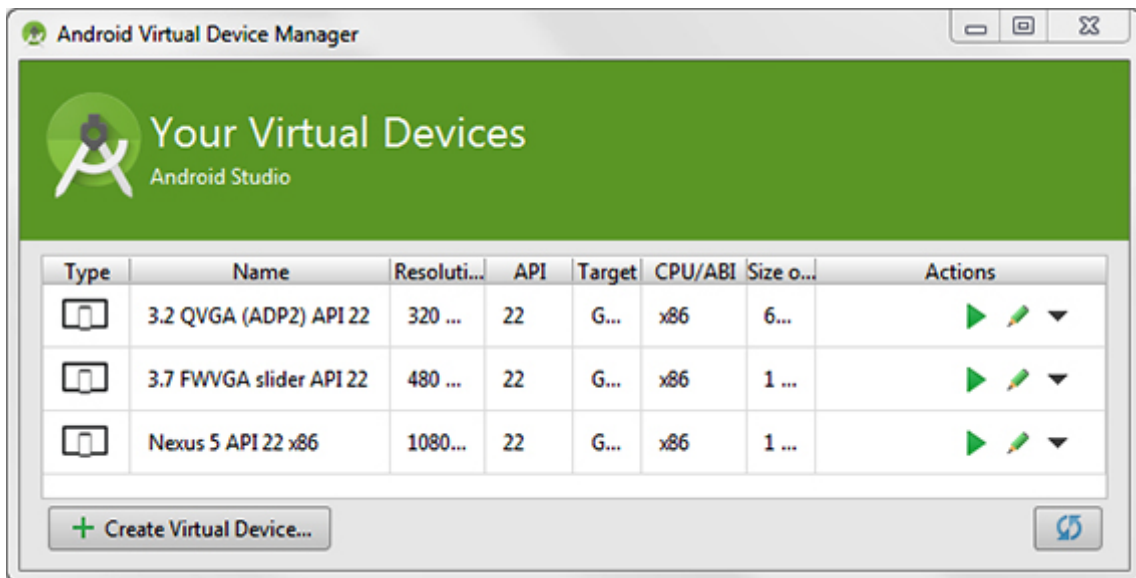
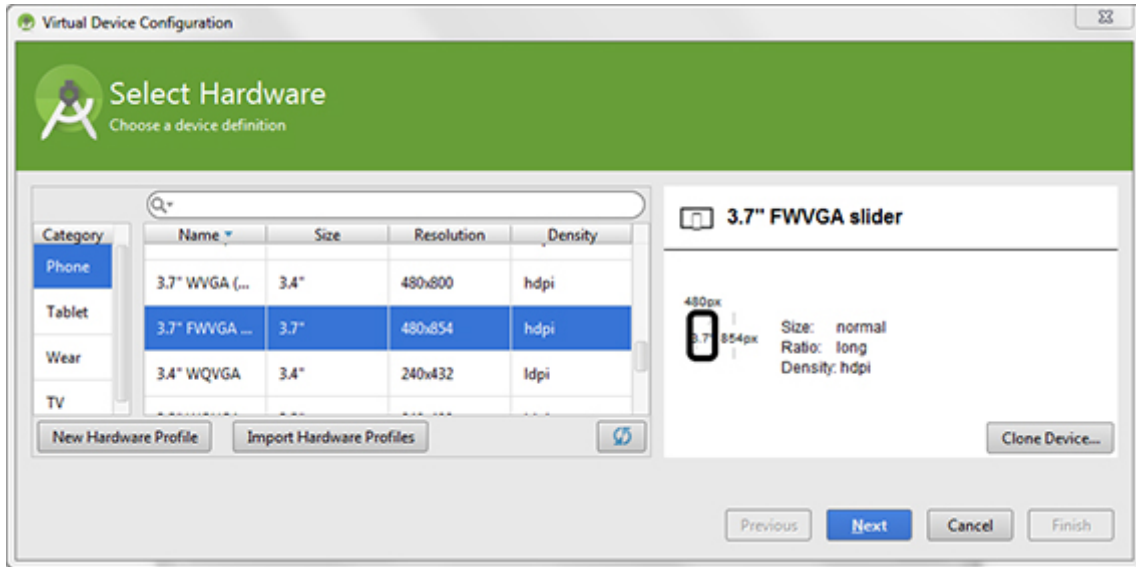
Build ⇒ Rebuild Project

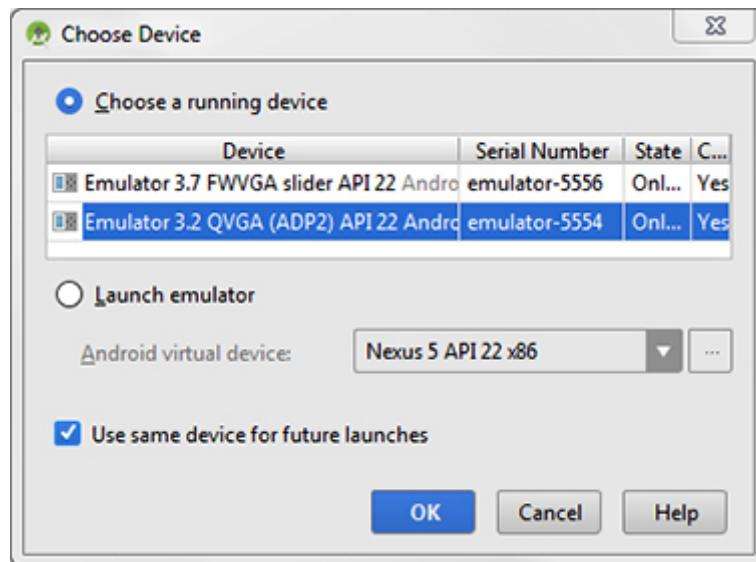


13. Run the App.

Before running the project, you may pick a running device by selecting the following options:

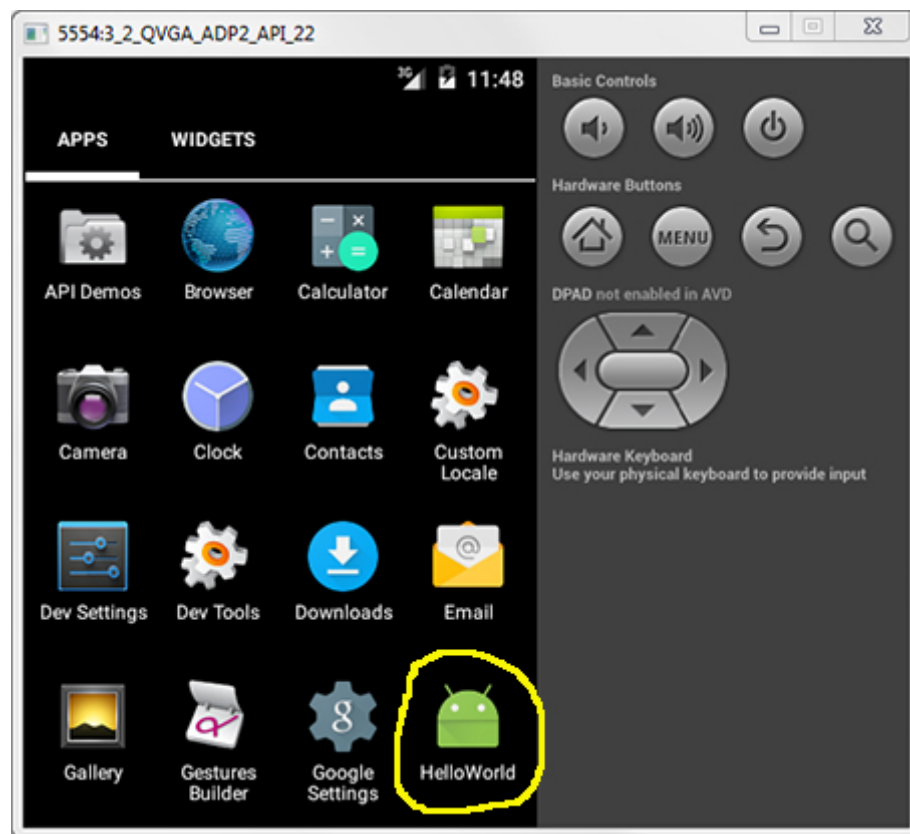
Tools ⇒ Android ⇒ AVD Manager

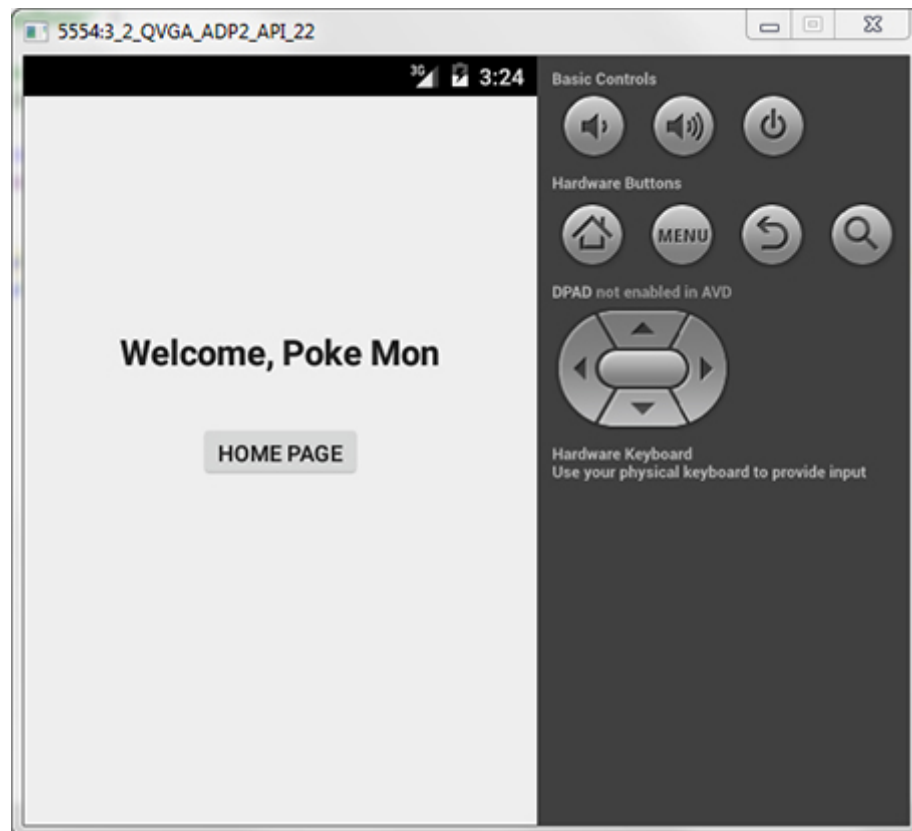
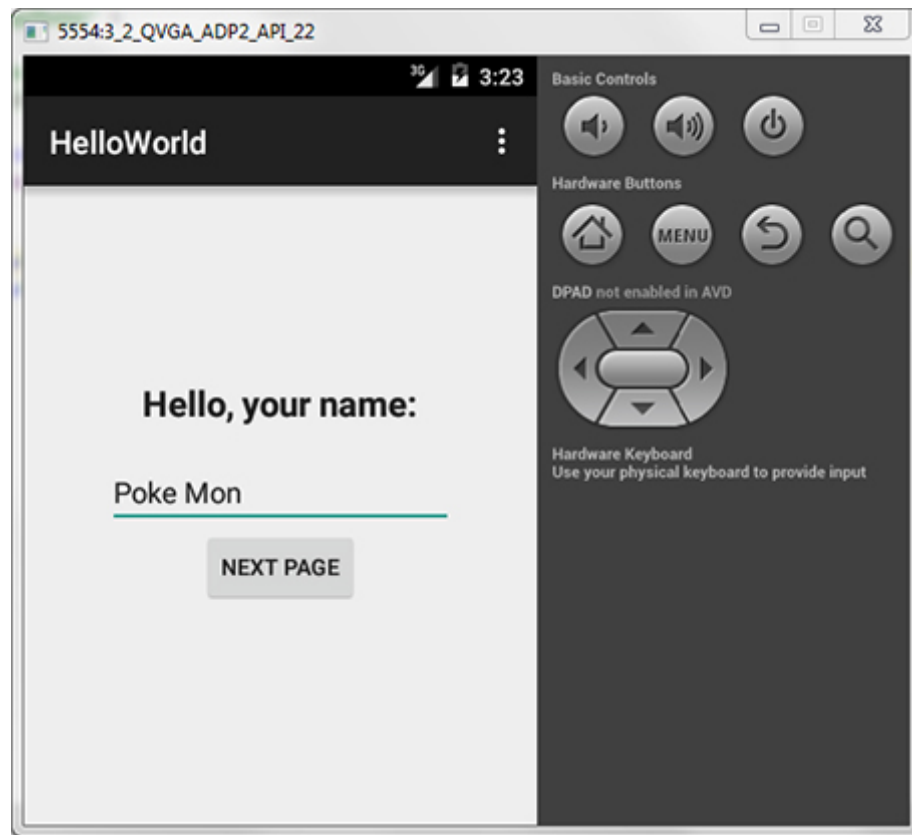


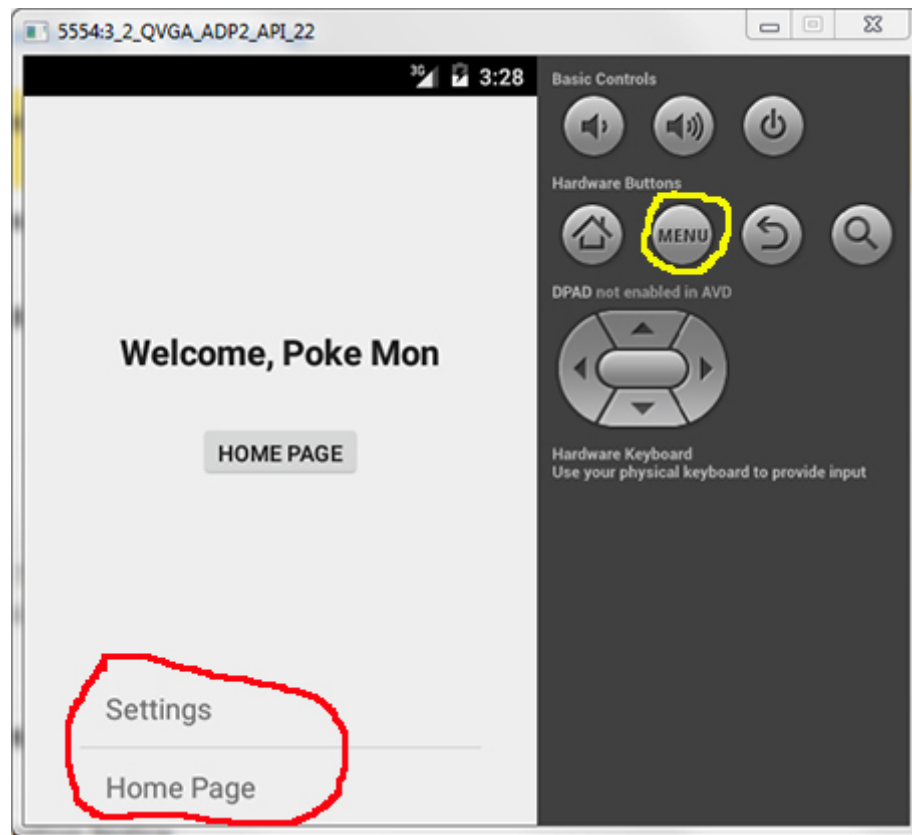


If the device does not start the project, run the project by selecting the following options:

Run ⇒ Run 'app'







Slide 7.3: Downloading and installing Android Studio

Slide 7.5: Android overview

Home

