[Tutorial #1] Environment setup: My First Android Project

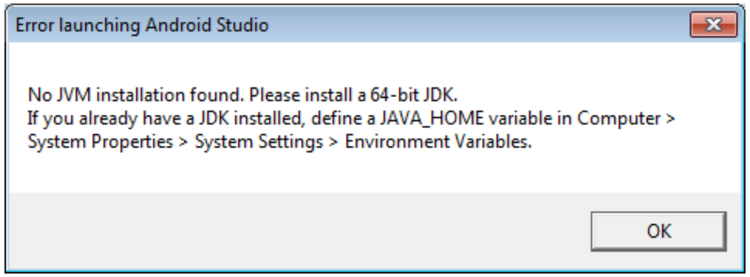
Android Developing - Environment Setup

Android Studio

- Download: http://developer.android.com/sdk/index.html
- Requirement:
  - Java JDK 7 or higher version
  - 2 GB memory
  - Windows / Mac OS X (10.8.5)

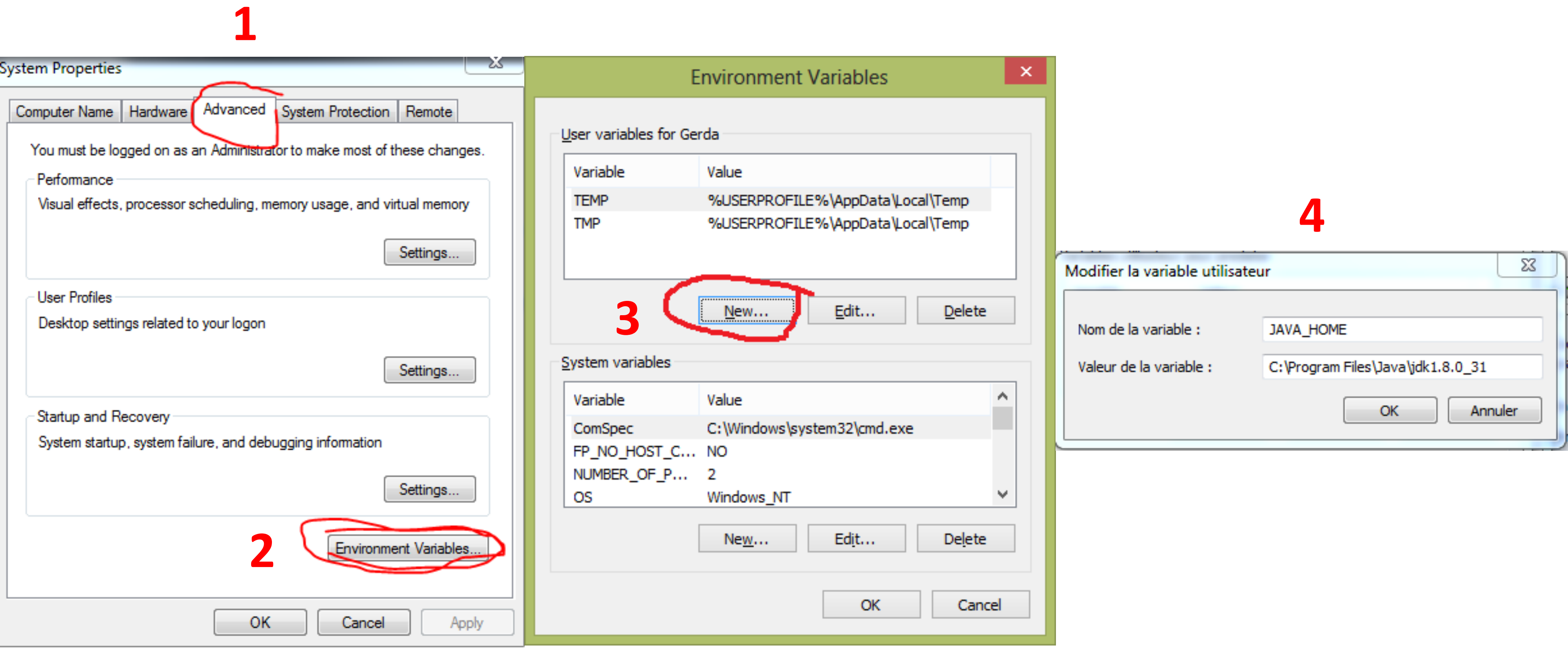
Error – Environment Variable

- Find the path of the installed Java JDK and add it as a system environment variable



Error launching Android Studio

No JVM installation found. Please install a 64-bit JDK.
If you already have a JDK installed, define a JAVA_HOME variable in Computer > System Properties > System Settings > Environment Variables.



You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your logon

Startup and Recovery
System startup, system failure, and debugging information

User variables for Gerda

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

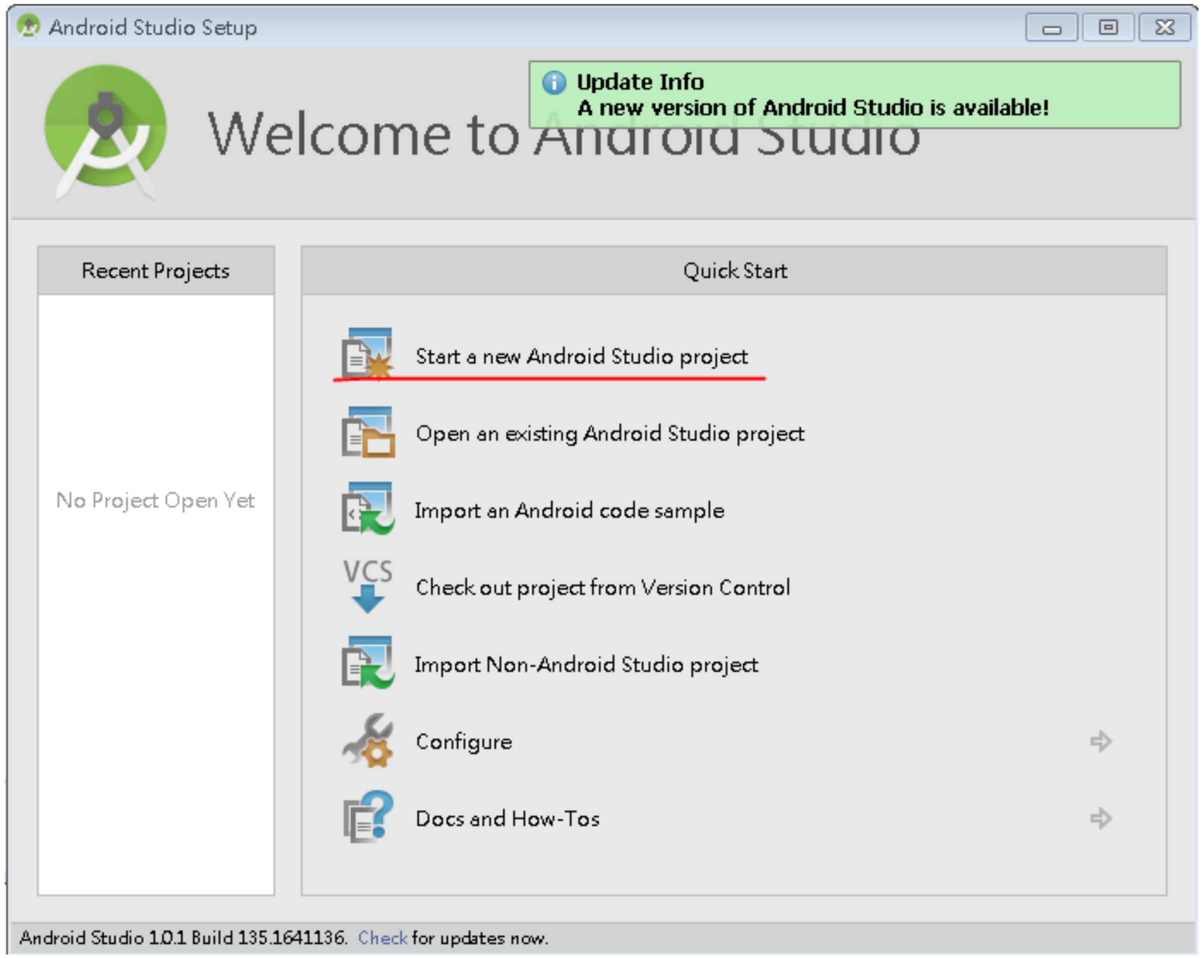
System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

Modifier la variable utilisateur

Nom de la variable : JAVA_HOME
Valeur de la variable : C:\Program Files\Java\jdk1.8.0_31

Create Your First Android Project



Android Studio Setup

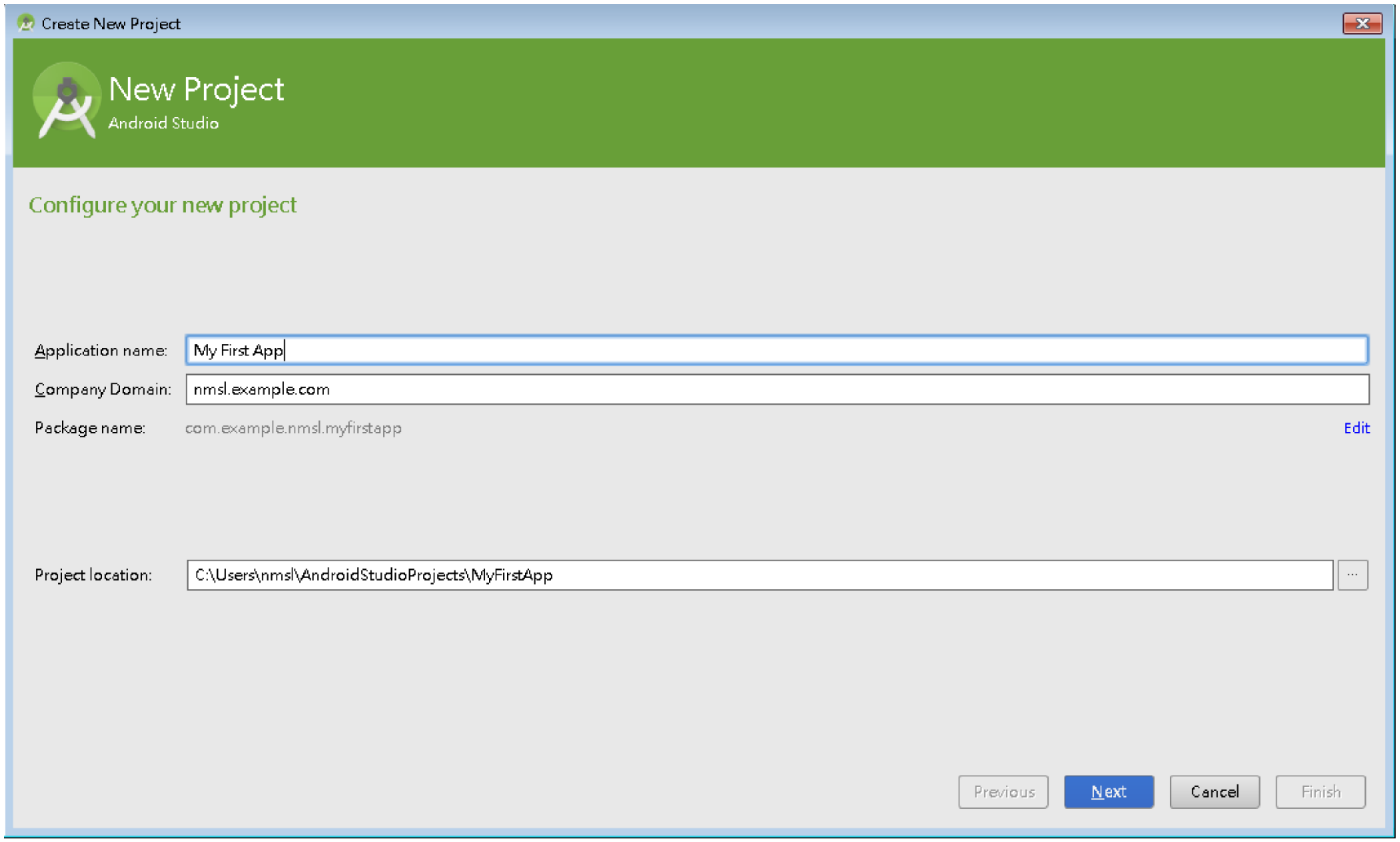
Update Info
A new version of Android Studio is available!

Welcome to Android Studio

Recent Projects
No Project Open Yet
Quick Start
Start a new Android Studio project
Open an existing Android Studio project
Import an Android code sample
Check out project from Version Control
Import Non-Android Studio project
Configure
Docs and How-Tos

Android Studio 1.0.1 Build 135.1641136. Check for updates now.

Your Project Name



Configure your new project

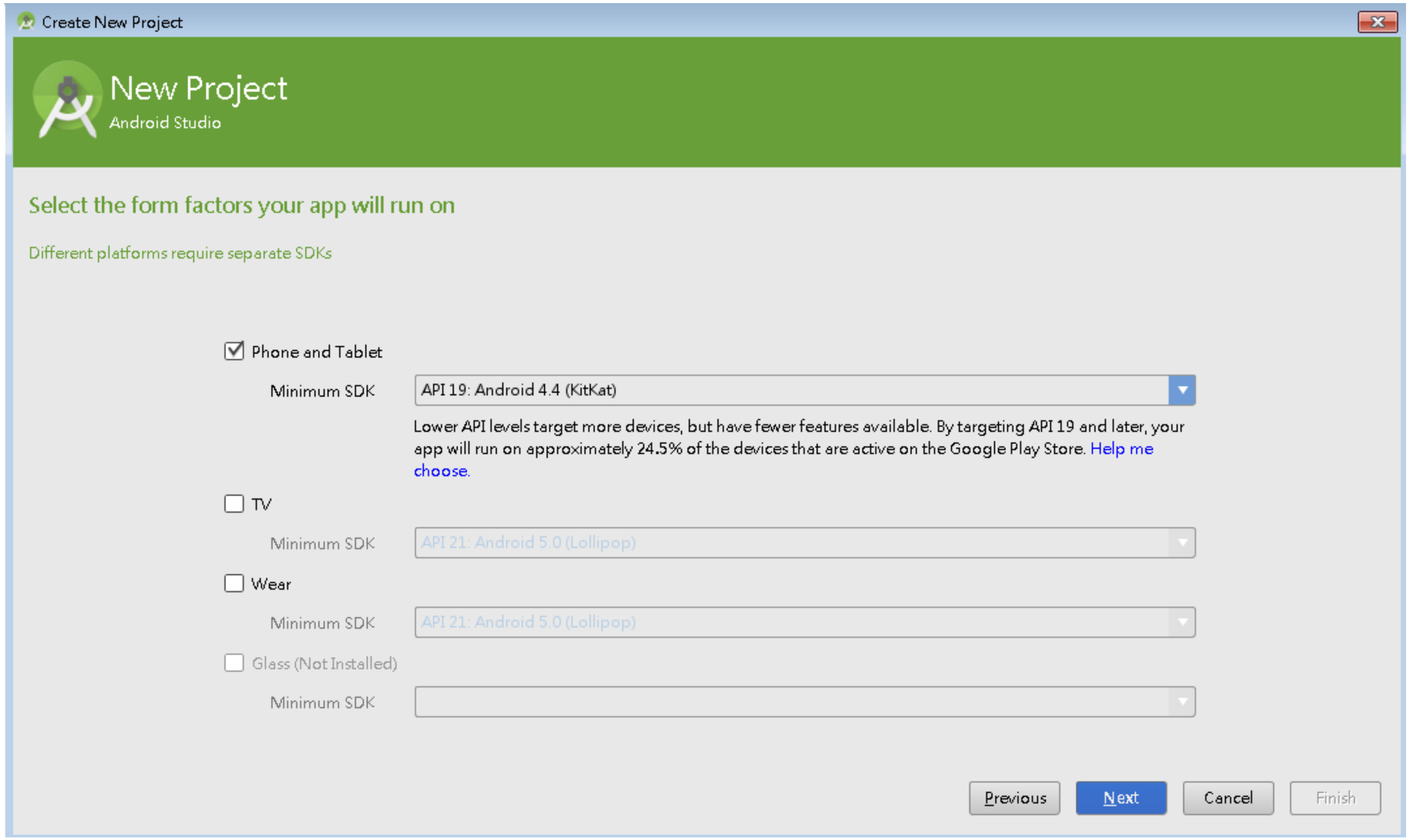
Application name: My First App

Company Domain: nmsl.example.com

Package name: com.example.nmsl.myfirstapp

Project location: C:\Users\nmsl\AndroidStudioProjects\MyFirstApp

Select the API Level



Select the form factors your app will run on

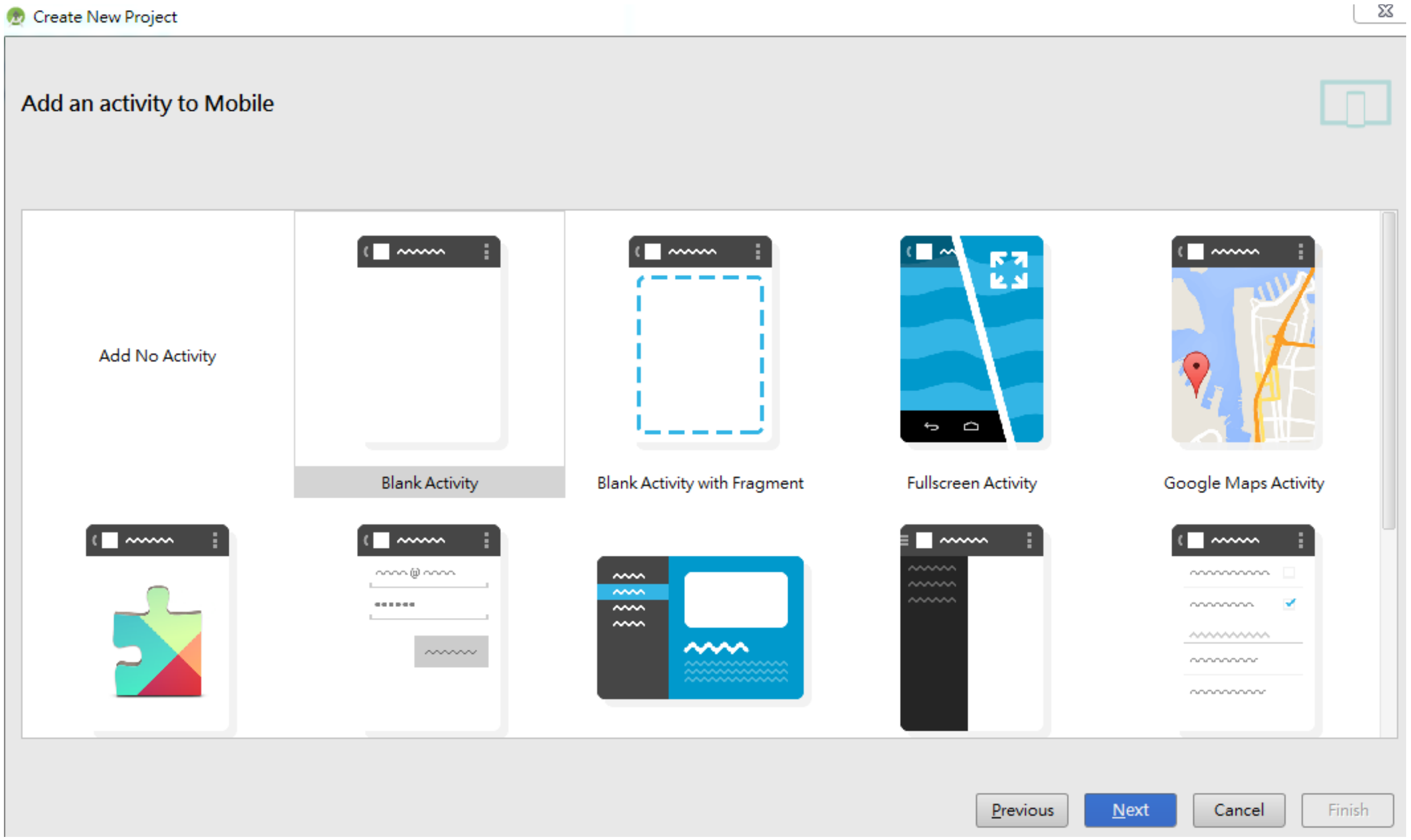
Different platforms require separate SDKs

- [✓] Phone and Tablet
  - Minimum SDK: API 19: Android 4.4 (KitKat)

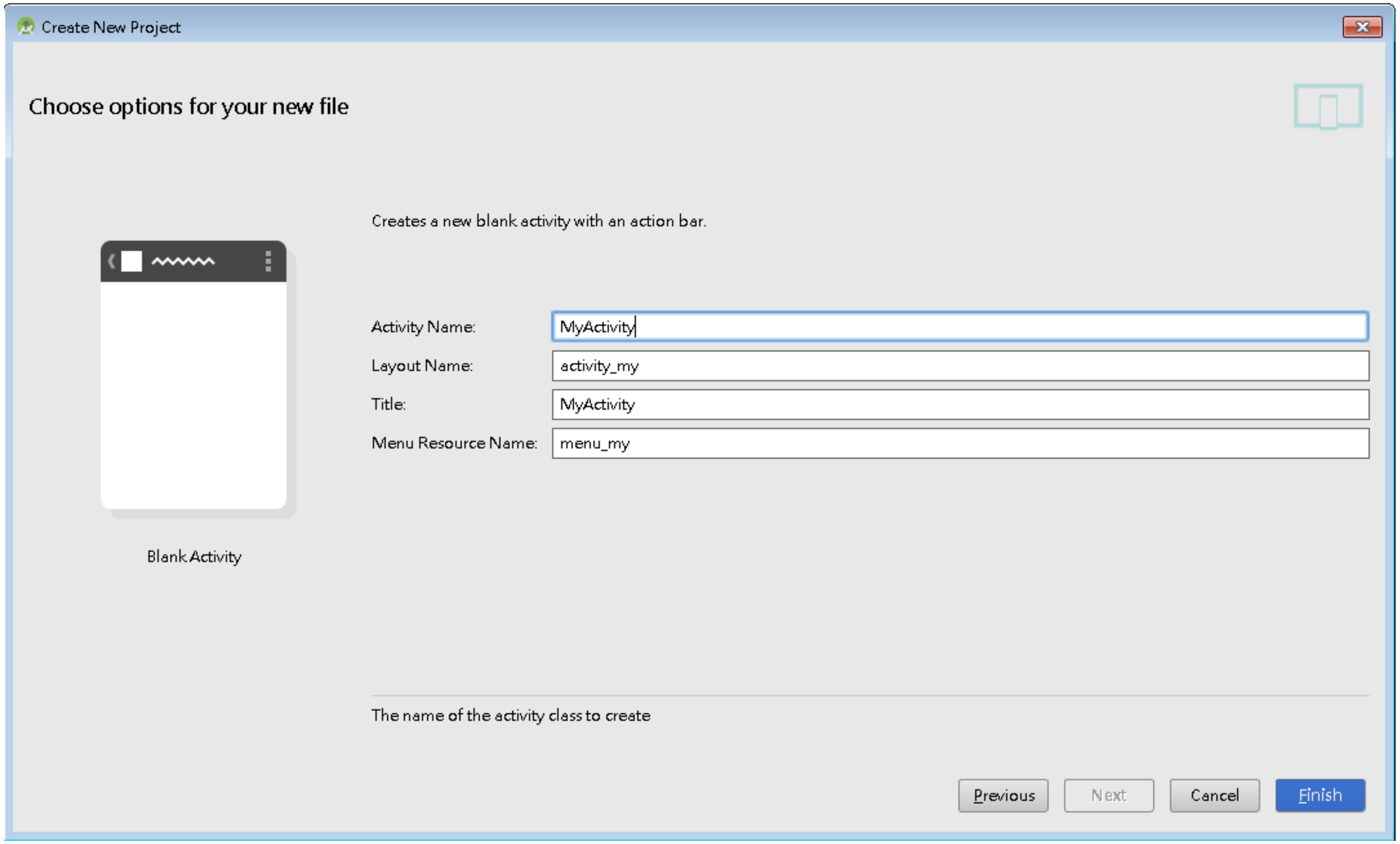
  Lower API levels target more devices, but have fewer features available. By targeting API 19 and later, your app will run on approximately 24.5% of the devices that are active on the Google Play Store. Help me choose.

- [ ] TV
  - Minimum SDK: API 21: Android 5.0 (Lollipop)
- [ ] Wear
  - Minimum SDK: API 21: Android 5.0 (Lollipop)
- [ ] Glass (Not Installed)
  - Minimum SDK:

Blank Activity



Your Activity Name



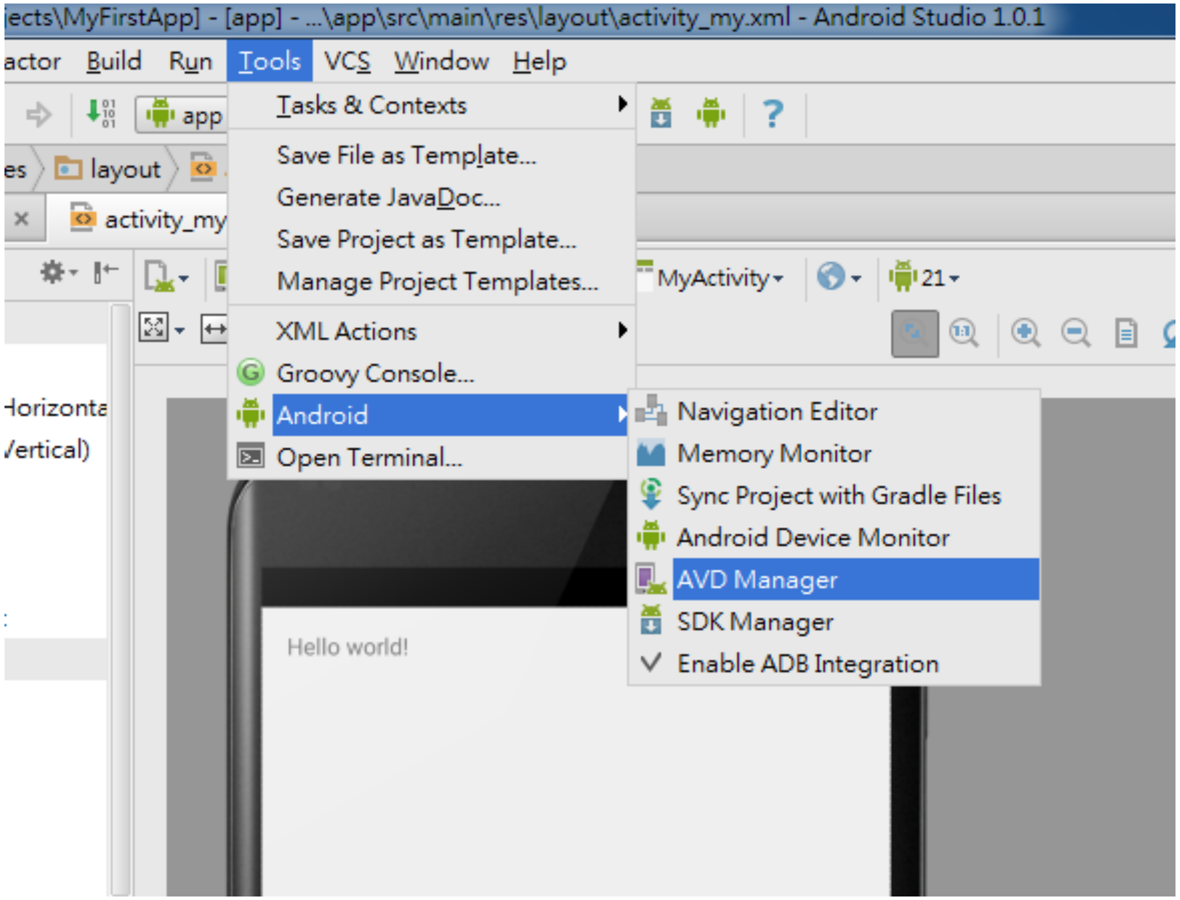
Choose options for your new file

Creates a new blank activity with an action bar.

Activity Name:	MyActivity
Layout Name:	activity_my
Title:	MyActivity
Menu Resource Name:	menu_my

The name of the activity class to create

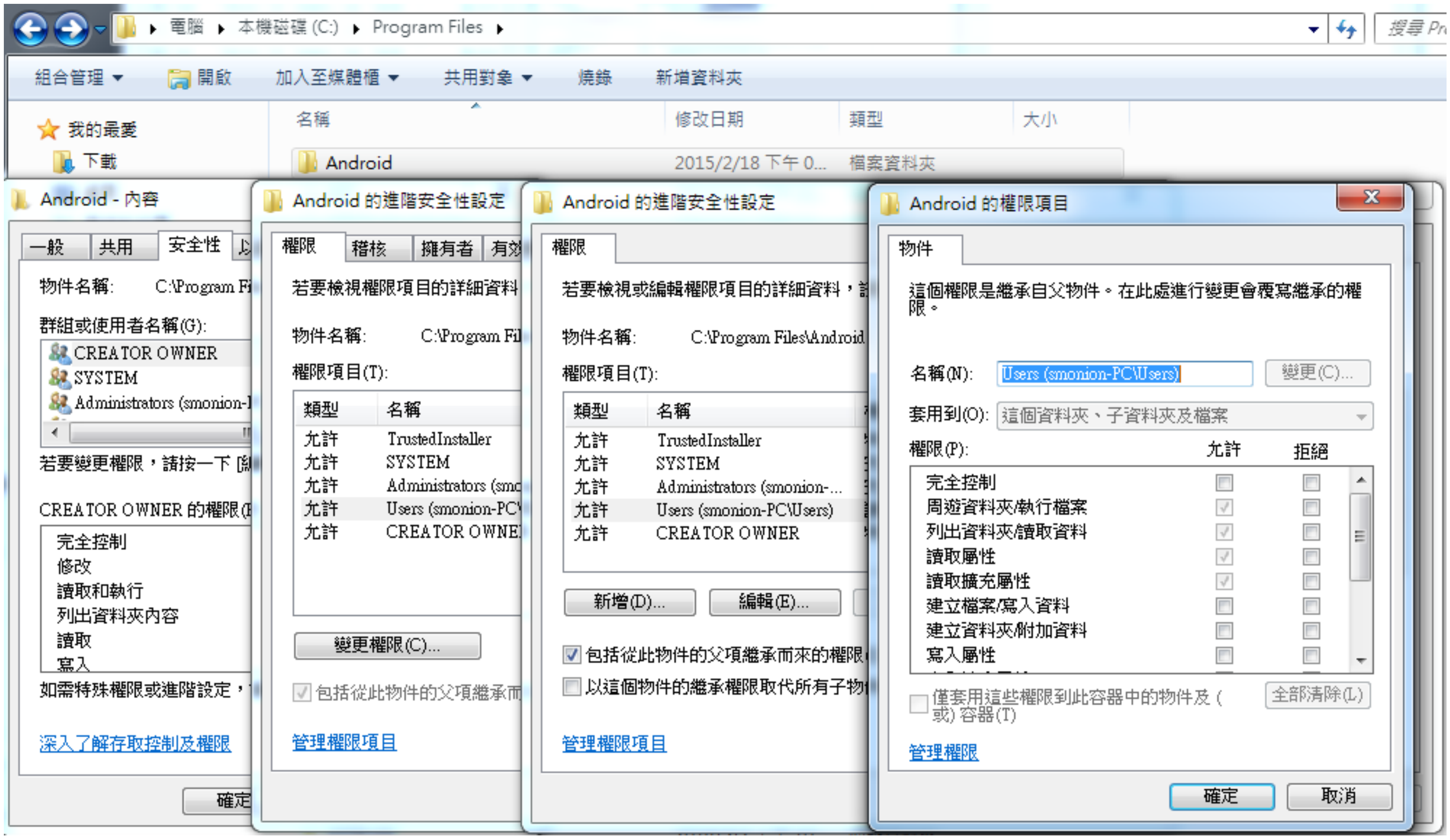
Create Your Emulator



Cannot See AVD Manager?

- If you cannot see the option of AVD Manager, please change the permission of your android studio folder

Modify Your Permission to Launch AVD Manager



電腦 ▸ 本機磁碟 (C:) ▸ Program Files ▸

組合管理 開啟 加入至媒體櫃 共用對象 燒錄 新增資料夾

我的最愛
下載

名稱	修改日期	類型	大小
Android	2015/2/18 下午 0...	檔案資料夾	

Android - 內容

一般 | 共用 | 安全性

物件名稱: C:\Program Fi

群組或使用者名稱(G):

- CREATOR OWNER
- SYSTEM
- Administrators (smonion-

若要變更權限，請按一下 [編

CREATOR OWNER 的權限(P)

- 完全控制
- 修改
- 讀取和執行
- 列出資料夾內容
- 讀取
- 寫入

如需特殊權限或進階設定，

深入了解存取控制及權限

確定

Android 的進階安全性設定

權限 | 稽核 | 擁有者 | 有效

若要檢視權限項目的詳細資料

物件名稱: C:\Program Fil

權限項目(T):

類型	名稱
允許	TrustedInstaller
允許	SYSTEM
允許	Administrators (smo
允許	Users (smonion-PC\
允許	CREATOR OWNE

變更權限(C)...

[✓] 包括從此物件的父項繼承而

管理權限項目

Android 的進階安全性設定

權限

若要檢視或編輯權限項目的詳細資料，

物件名稱: C:\Program Files\Android

權限項目(T):

類型	名稱
允許	TrustedInstaller
允許	SYSTEM
允許	Administrators (smonion-...
允許	Users (smonion-PC\Users)
允許	CREATOR OWNER

新增(D)... 編輯(E)...

[✓] 包括從此物件的父項繼承而來的權限

[ ] 以這個物件的繼承權限取代所有子物

管理權限項目

Android 的權限項目

物件

這個權限是繼承自父物件。在此處進行變更會覆寫繼承的權限。

名稱(N): Users (smonion-PC\Users) 變更(C)...

套用到(O): 這個資料夾、子資料夾及檔案

權限(P):	允許	拒絕
完全控制	☐	☐
周遊資料夾/執行檔案	✓	☐
列出資料夾/讀取資料	✓	☐
讀取屬性	✓	☐
讀取擴充屬性	✓	☐
建立檔案/寫入資料	☐	☐
建立資料夾/附加資料	☐	☐
寫入屬性	☐	☐

[ ] 僅套用這些權限到此容器中的物件及(或)容器(T)

全部清除(L)

管理權限

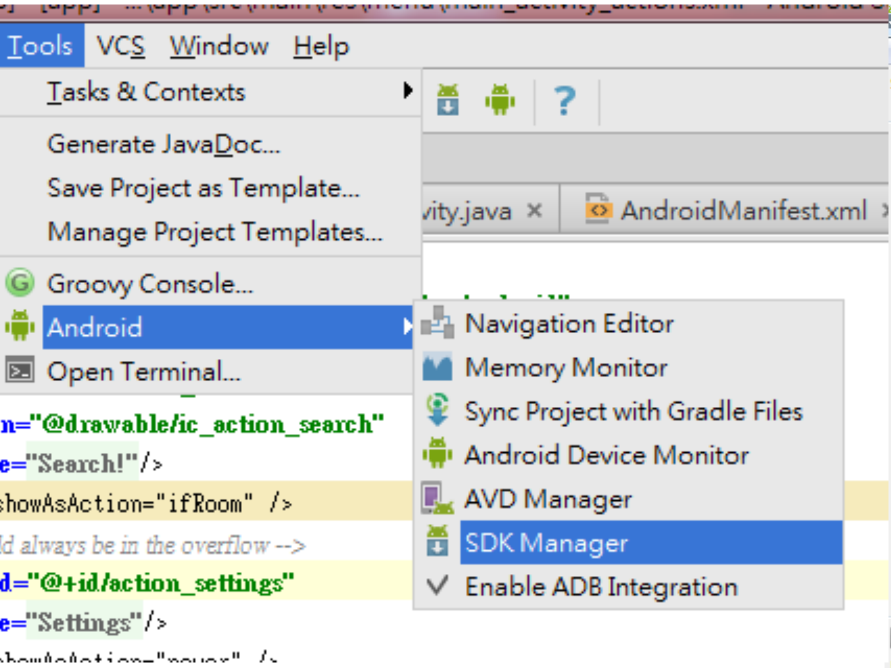
確定 取消

Launch the Default Emulator

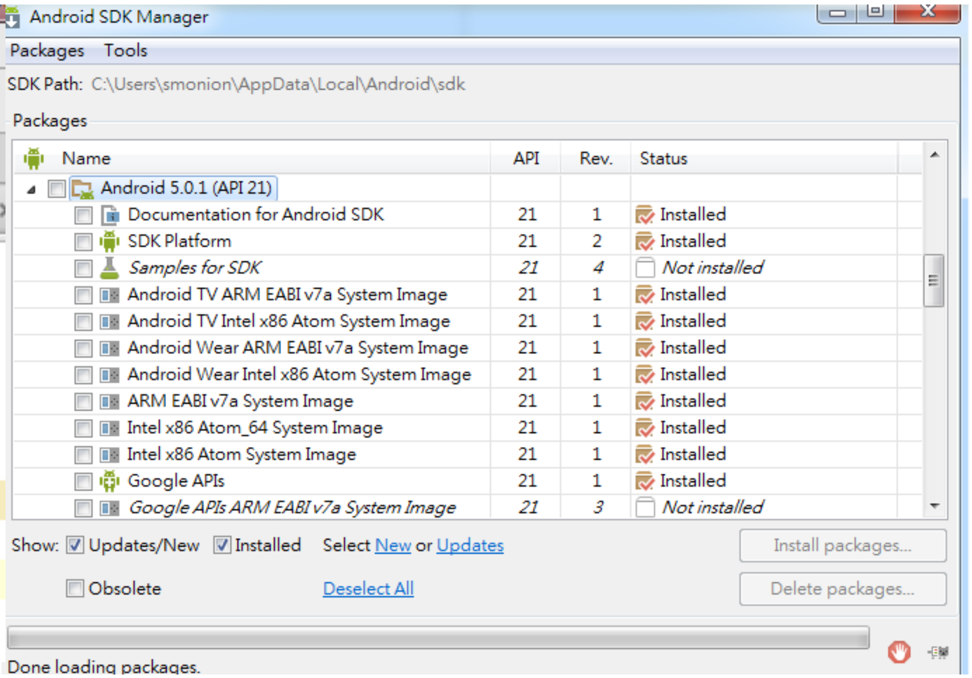
- Please run the default Nexus 5 Emulator using AVD Manager
- If you would like to create your own emulator, you need to update your SDK packages first.

SDK Manager

- Update your SDK package using SDK Manager



```
n="@drawable/ic_action_search"
e="Search!"/>
showAsAction="ifRoom" />
ld always be in the overflow -->
d="@+id/action_settings"
e="Settings"/>
```



SDK Path: C:\Users\smonion\AppData\Local\Android\sdk

Packages

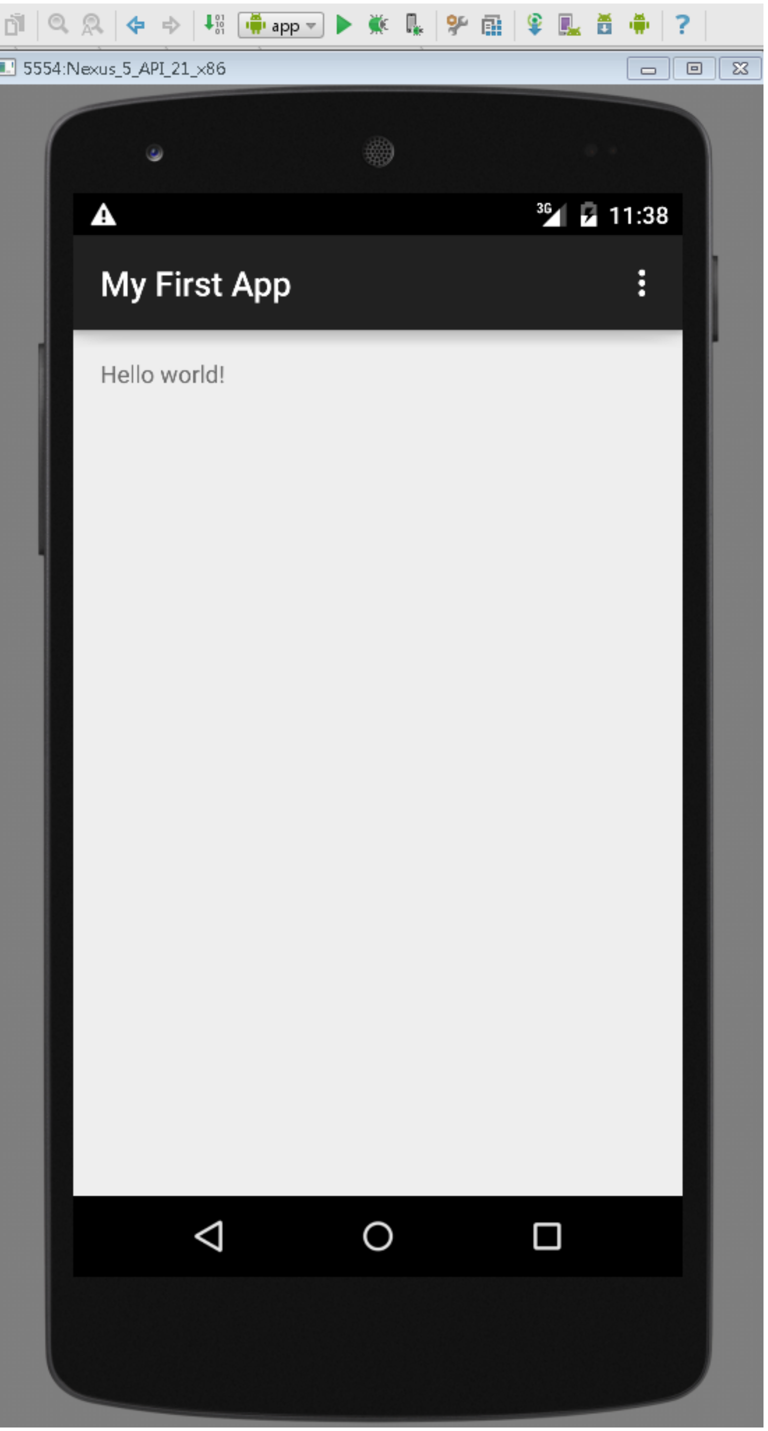
Name	API	Rev.	Status
Android 5.0.1 (API 21)			
Documentation for Android SDK	21	1	Installed
SDK Platform	21	2	Installed
Samples for SDK	*21*	*4*	*Not installed*
Android TV ARM EABI v7a System Image	21	1	Installed
Android TV Intel x86 Atom System Image	21	1	Installed
Android Wear ARM EABI v7a System Image	21	1	Installed
Android Wear Intel x86 Atom System Image	21	1	Installed
ARM EABI v7a System Image	21	1	Installed
Intel x86 Atom_64 System Image	21	1	Installed
Intel x86 Atom System Image	21	1	Installed
Google APIs	21	1	Installed
Google APIs ARM EABI v7a System Image	*21*	*3*	*Not installed*

Show: [✓ Updates/New] [✓ Installed] Select New or Updates

Obsolete

Deselect All

Done loading packages.



- Compile and run your project and you can see the message on your virtual Nexus 5!

Your First App

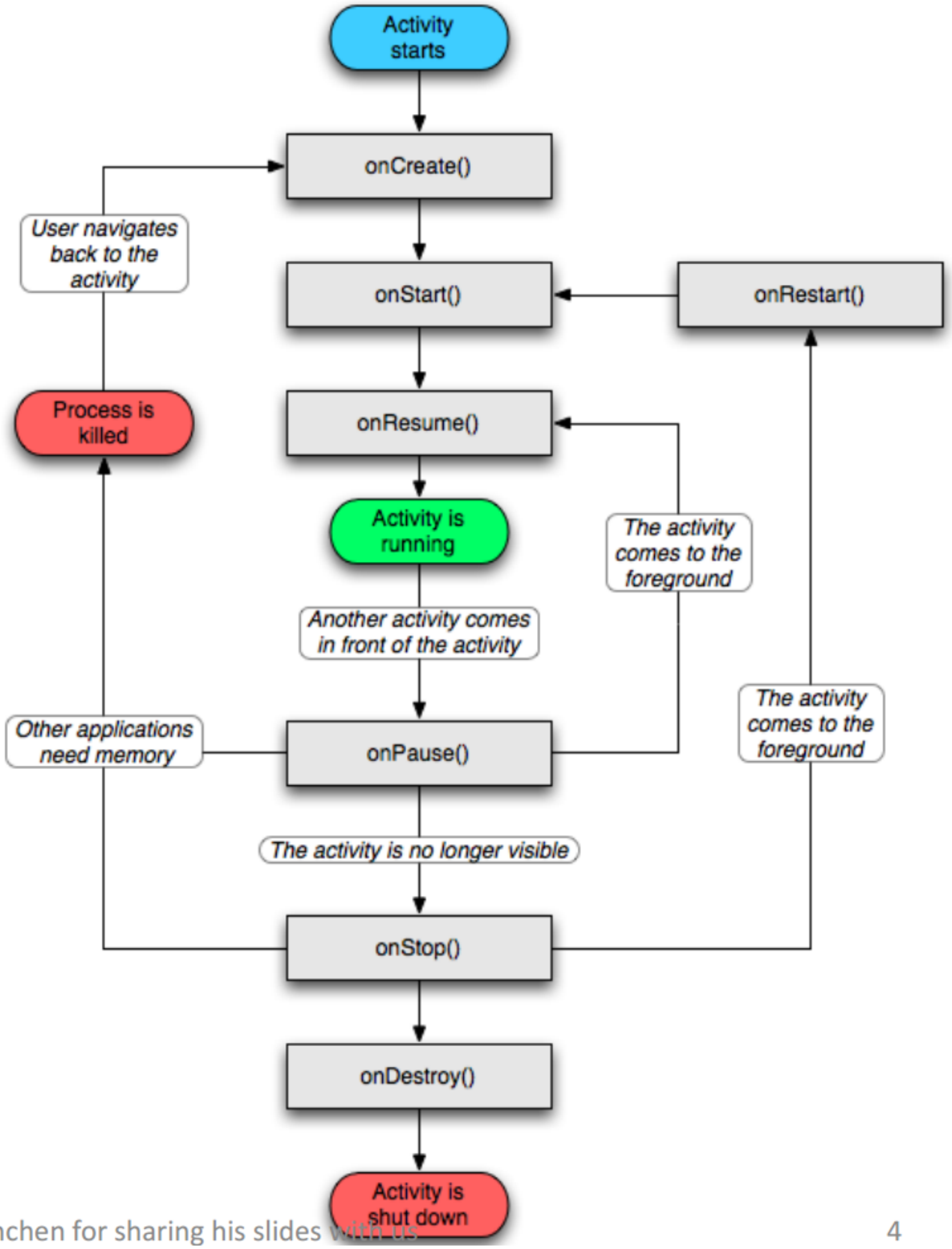
Android Activity

- Interact with users
- Visual User interface
- Hierarchy of views
- One or several activities in an application

Activity Life Cycle

- Activities are managed as an activity stack. When a new activity is started, it is placed on the top of the stack and becomes the running activity
- States:
  - Active / running: activity in the foreground
  - Pause: An activity has lost focus but is still visible
  - Stopped: It's no longer visible but still retains all state member information
  - Finish / Kill

Android Life Cycle


Activity starts
onCreate()
User navigates back to the activity
onStart()
onRestart()
Process is killed
onResume()
Activity is running
The activity comes to the foreground
Another activity comes in front of the activity
Other applications need memory
onPause()
The activity comes to the foreground
The activity is no longer visible
onStop()
onDestroy()
Activity is shut down

Android Manifest

- The components used in an android application should be declared
  - Activity
  - Intent
  - …

Manifest Example

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

  package="com.example.nmsl.myfirstapp" >

  <application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
      android:name=".MainActivity"
      android:label="@string/app_name" >
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>

</manifest>
```

The Package Name

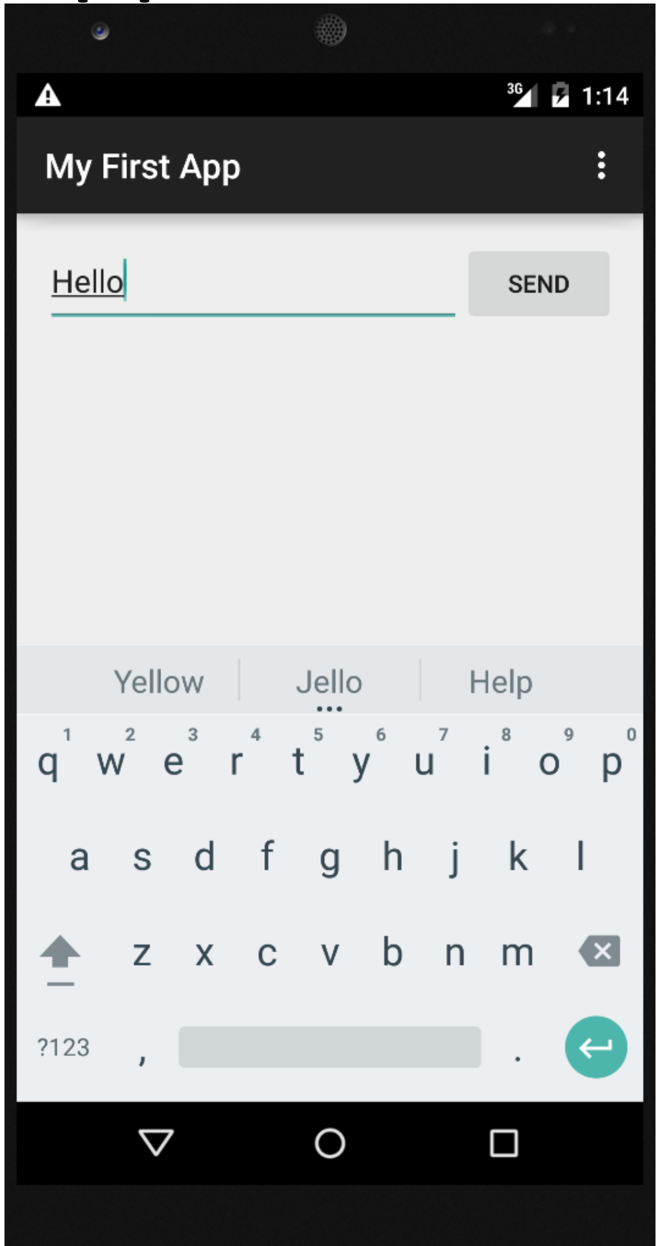
Describe Your Application

Activity Component

Intent Component

Your First App

- Edit Text
- Button
  - Listener
  - Send Message
  - Create Second Activity



Steps

- Create a linear layout
- Add your view objects into the layout
- Create the resources used in the view objects
- Create the function to do interaction while we push the button

Linear Layout

- Edit your activity_main.xml

```
<LinearLayout 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_ma
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity"
  android:orientation="horizontal">

  <EditText android:id="@+id/edit_message"
    android:layout_weight="1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="@string/edit_message" />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send“
</LinearLayout>
```

The default value of weight of each view is 0

Missing the String

The width and height can just contain the view

Missing the String

Add String Resources

- Edit string.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">My First App</string>
    <string name="edit_message">Enter a message</string>
    <string name="button_send">Send</string>
    <string name="action_settings">Settings</string>
    <string name="title_activity_main">MainActivity</string>
</resources>
```

Starting Another Activity

- Link your button with a function to do something
- Edit activity_main.xml

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send"
    android:onClick="sendMessage" />
```

The name of your function

- Edit MyActivity.java to add the function

Create the function

```
import android.content.Intent;
…
public void sendMessage(View view) {
      // Do something in response to button
      Intent intent = new Intent(this, DisplayMessageActivity.class);
      EditText editText = (EditText) findViewById(R.id.edit_message);
      String message = editText.getText().toString();
      intent.putExtra(EXTRA_MESSAGE, message);
      startActivity(intent);
   }
```

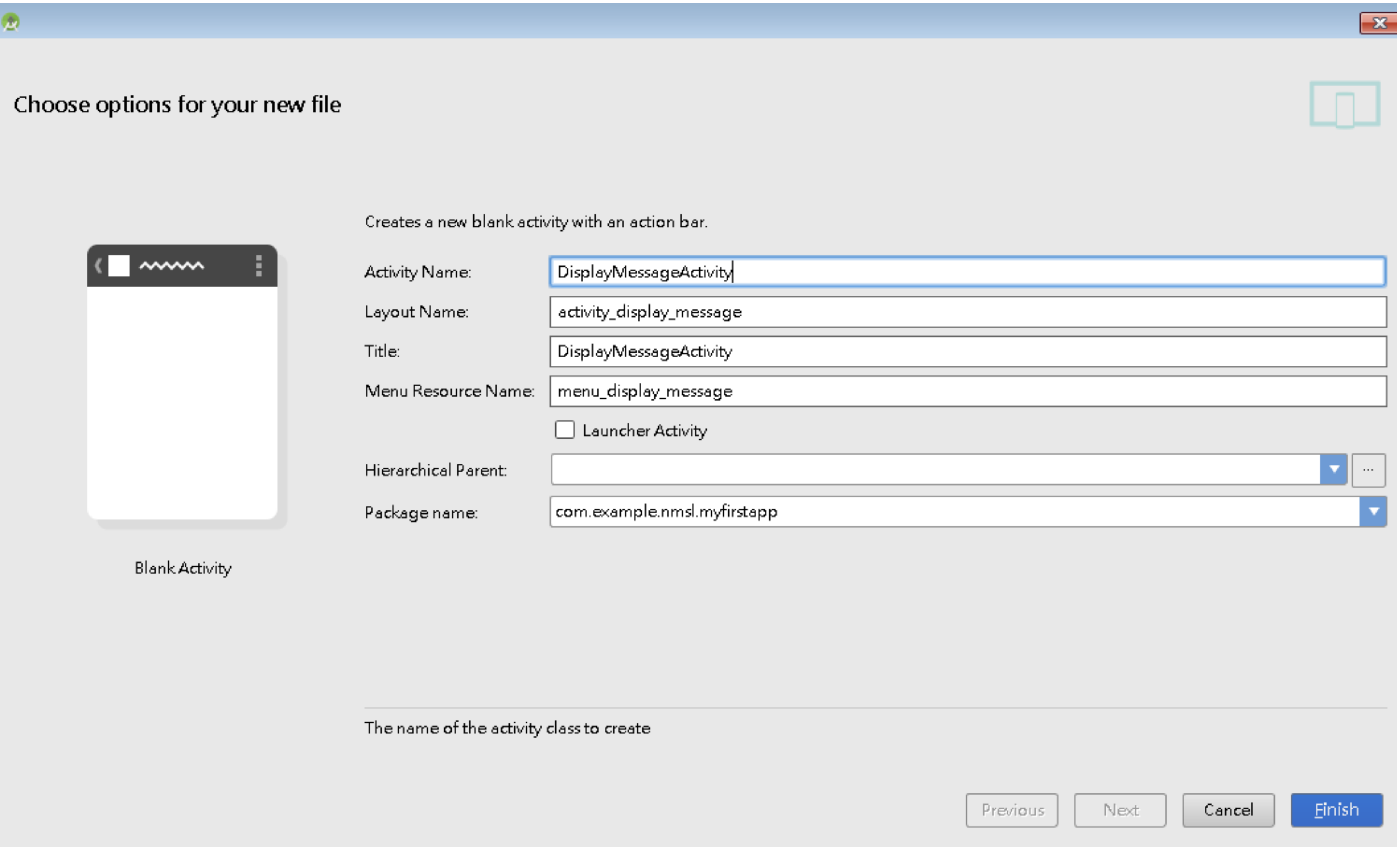
Create a Unique Key

```
public void sendMessage(View view) {
        // Do something in response to button
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        EditText editText = (EditText) findViewById(R.id.edit_message);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }


public class MyActivity extends ActionBarActivity {
    public final static String EXTRA_MESSAGE = "com.mycompany.myfirstapp.MESSAGE";
    ...
}
```

Create a unique key for the message put by the intent. We then get the message by this key in the second activity (next page)

Create the Second Activity


Choose options for your new file

Creates a new blank activity with an action bar.

Activity Name:	DisplayMessageActivity
Layout Name:	activity_display_message
Title:	DisplayMessageActivity
Menu Resource Name:	menu_display_message
	☐ Launcher Activity
Hierarchical Parent:	
Package name:	com.example.nmsl.myfirstapp

The name of the activity class to create

Add String Resources

- Check you manifest. You can see a new activity and it needs one more string resource

```
<resources>
   <string name="app_name">My First App</string>
   <string name="edit_message">Enter a message</string>
   <string name="action_settings">Settings</string>
   <string name="button_send">Send</string>
   <string name="title_activity_display_message">My Message</string>
   <string name="hello_world">Hello world!</string>
   <string name="action_search">Search!</string>
</resources>
```

Delete Unused View Object

```
<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_ma
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin"
   tools:context="com.example.nmsl.myfirstapp.DisplayMessageActivity">

   <TextView android:text="@string/hello_world" android:layout_width="wrap_content"
      android:layout_height="wrap_content" />

</RelativeLayout>
```

Delete it
We do not need default text view in second activity

Receive the Intent

- Edit DisplayMessageActivity.java
  - Get the message from the intent
  - Create a textview to show the message

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Intent intent = getIntent();
        String message = intent.getStringExtra(MyActivity.EXTRA_MESSAGE);
        TextView textView = new TextView(this);
        textView.setTextSize(40);
        textView.setText(message);
        setContentView(textView);
    }
```