Android Development Tool

Mr.Yous Sopheap

Course Overview

1. Android environment setup
2. Create New Project

System Requirements

- Hardware
  - Windows 7,8,10 (32- or 64-bit)
  - Mac OS X 10.5.8 or later (x86 only)
  - Linux (Ubuntu, Lucid Lynx)
- Java Platform, SE
  - JDK 6 or 7
    - Java 8 features are not supported
    - JRE alone is not sufficient
  - Install Java SE first.
    - Set environment variable JAVA_HOME

I-Android Environment Setup

1. Setup Java Development Kit (JDK)
2. Set Your JAVA_HOME
3. Install Android Studio
4. Check for Update
5. Install System Images and Tools in SDK Manager
6. Create Android Virtual Device

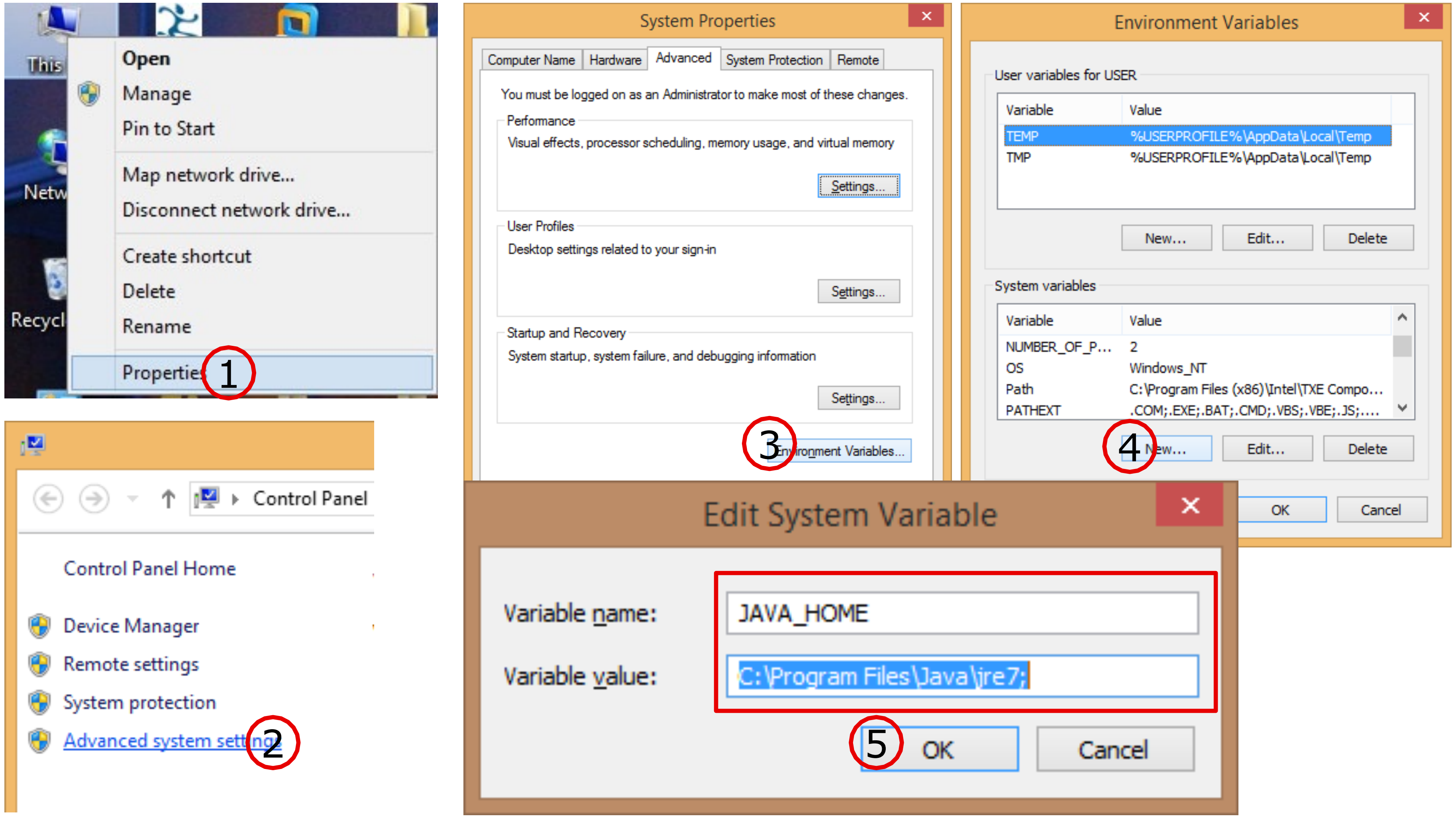
1-Setup JDK

1. check if you have the Java Developer Kit (JDK) version 6.0 or greater already installed
2. If you do not have Java installed, or if your version is below 6.0, install the Java JDK

2- Set Your JAVA_HOME

1. Right click on the Computer- Properties
2. Advanced System Settings
3. In the Advanced tab-Click Environment Variables
4. Create System Variable by Click New Button
5. Set:
    1. Variable Name: JAVA_HOME
    2. Variable Value: C:\Program Files\Java\jre7;
6. Click Button OK-OK-OK

Practice



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

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

Desktop settings related to your sign-in

System startup, system failure, and debugging information

User variables for USER

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

System variables

Variable	Value
NUMBER_OF_P...	2
OS	Windows_NT
Path	C:\Program Files (x86)\Intel\TXE Compo...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jre7;

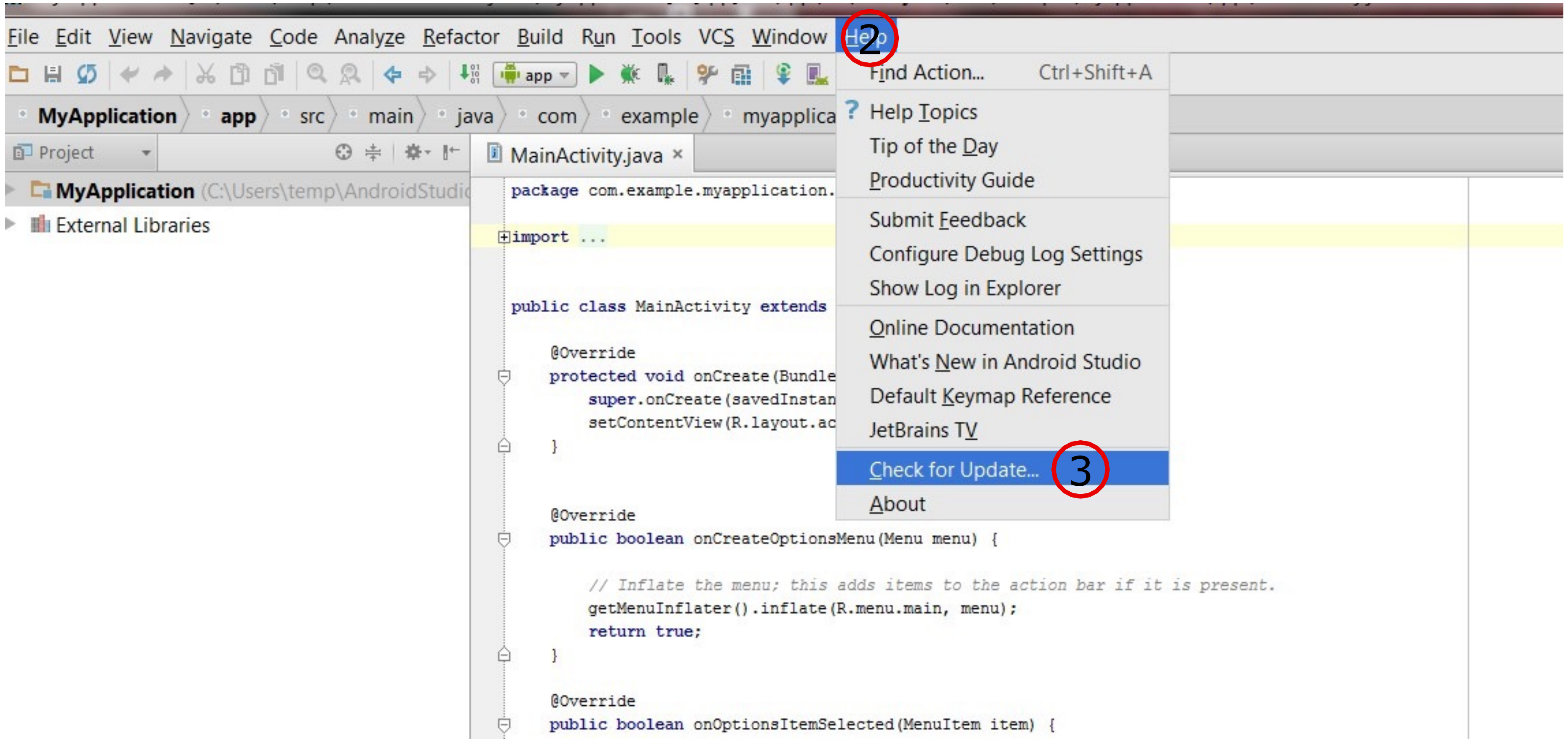
3- Install Android Studio

1. Download: http://developer.android.com/sdk/installing/ studio.html
2. Open the downloaded file, and follow the Android Studio Setup Wizard to complete the installation.

4- Check for Update

1. Open Android Studio.
2. click “Check for updates now” on the green pop-up alerting you of new updates or navigate to it from the Help menu-Check for update.
3. After updates are installed and then restart Android Studio.

Practice

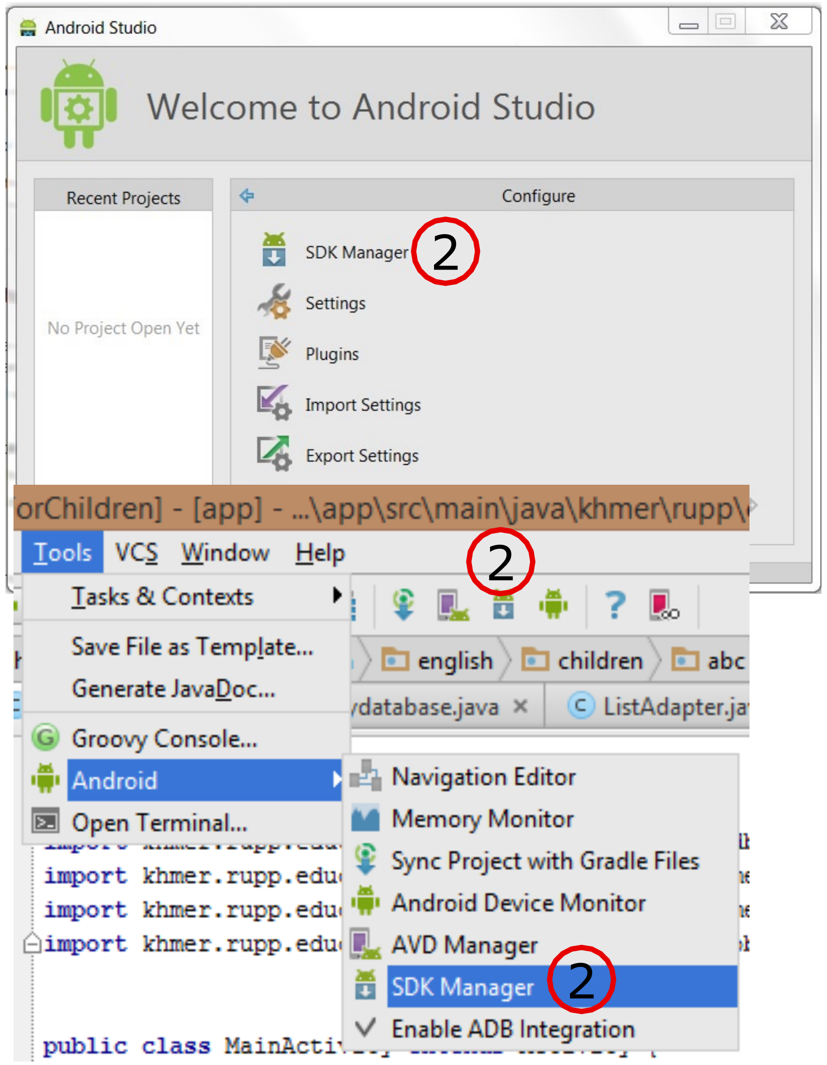

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
2
app
MyApplication app src main java com example myapplica
Project
MainActivity.java
MyApplication (C:\Users\temp\AndroidStudi
External Libraries
Find Action... Ctrl+Shift+A
Help Topics
Tip of the Day
Productivity Guide
Submit Feedback
Configure Debug Log Settings
Show Log in Explorer
Online Documentation
What's New in Android Studio
Default Keymap Reference
JetBrains TV
Check for Update...
3
About

```
package com.example.myapplication.

import ...


public class MainActivity extends

    @Override
    protected void onCreate(Bundle
        super.onCreate(savedInstan
        setContentView(R.layout.ac
    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
```

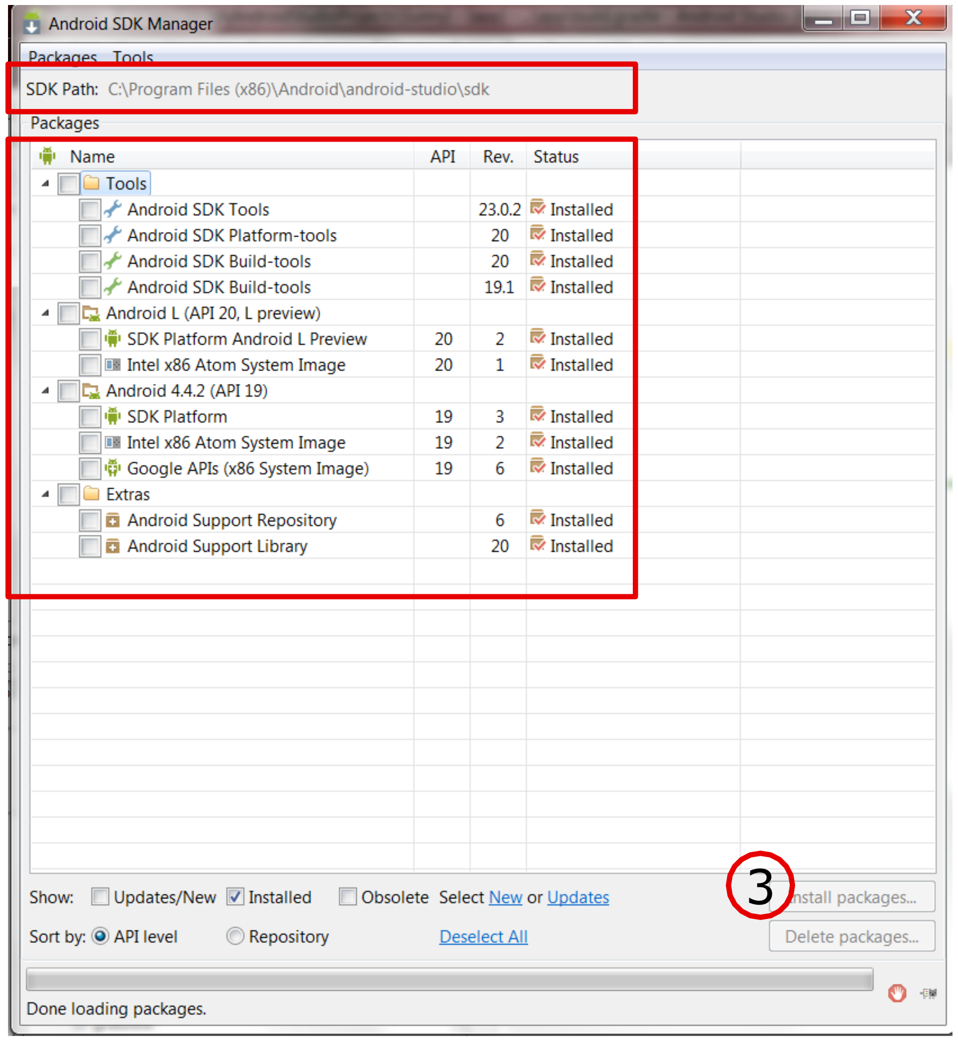
5- Install System Images and Tools in SDK Manager

1. Launch Android Studio
2. You should see a “Welcome to Android Studio” window. Select **Configure → SDK Manager**.
3. From the SDK manager, you will see a lot of different packages that you can install. SELECT ONLY THE BOXES SHOWN BELOW
4. Accept the license for each set of packages in order to complete the install. You may need to install each set of packages separately, as they fall under separate licenses.

Practice



```
import khmer.rupp.edu
import khmer.rupp.edu
import khmer.rupp.edu

public class MainActi
```



SDK Path: C:\Program Files (x86)\Android\android-studio\sdk

Name	API	Rev.	Status
Tools			
Android SDK Tools		23.0.2	Installed
Android SDK Platform-tools		20	Installed
Android SDK Build-tools		20	Installed
Android SDK Build-tools		19.1	Installed
Android L (API 20, L preview)			
SDK Platform Android L Preview	20	2	Installed
Intel x86 Atom System Image	20	1	Installed
Android 4.4.2 (API 19)			
SDK Platform	19	3	Installed
Intel x86 Atom System Image	19	2	Installed
Google APIs (x86 System Image)	19	6	Installed
Extras			
Android Support Repository		6	Installed
Android Support Library		20	Installed

Show: ☐ Updates/New ☑ Installed ☐ Obsolete Select New or Updates

Sort by: ● API level ○ Repository Deselect All

Done loading packages.

6- Android Virtual Device (AVD)

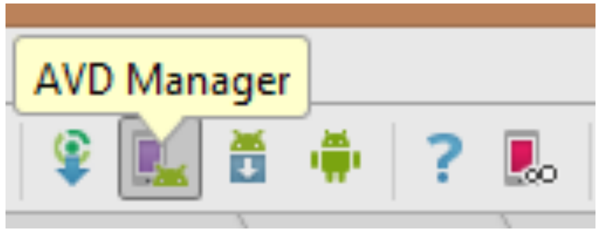
- An emulator configuration to simulate an actual device
  - define hardware and software options
- An AVD consists of
  - A hardware profile, e.g., keyboard, memory, etc.
  - A system image: target CPU chipset, API level
  - Other options: skins, appearances, SD cards, etc.
- You need at least one AVD that
  - AVD API level $\geq$ Minimum API level of your app

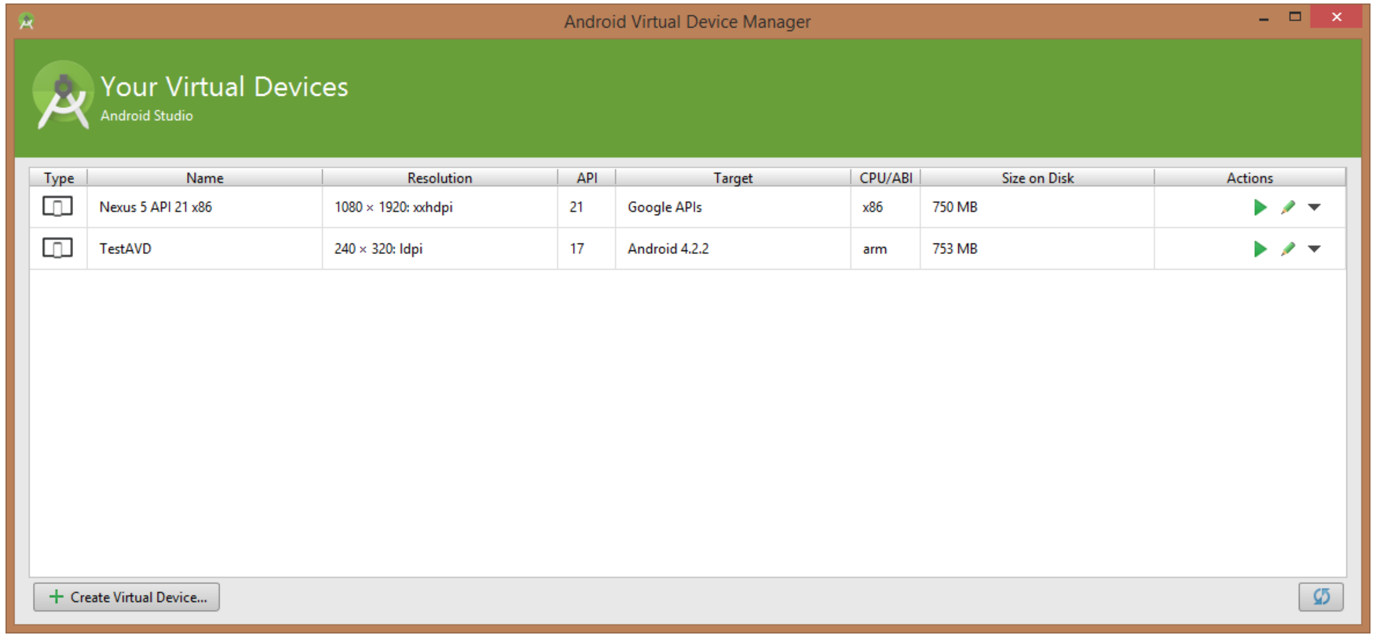
Launch AVD Manager

- In Android Studio
  - **Tools | Android | AVD Manager,**

    or
  - the AVD Manager icon in the toolbar.
- Command line:

  android avd




Android Virtual Device Manager

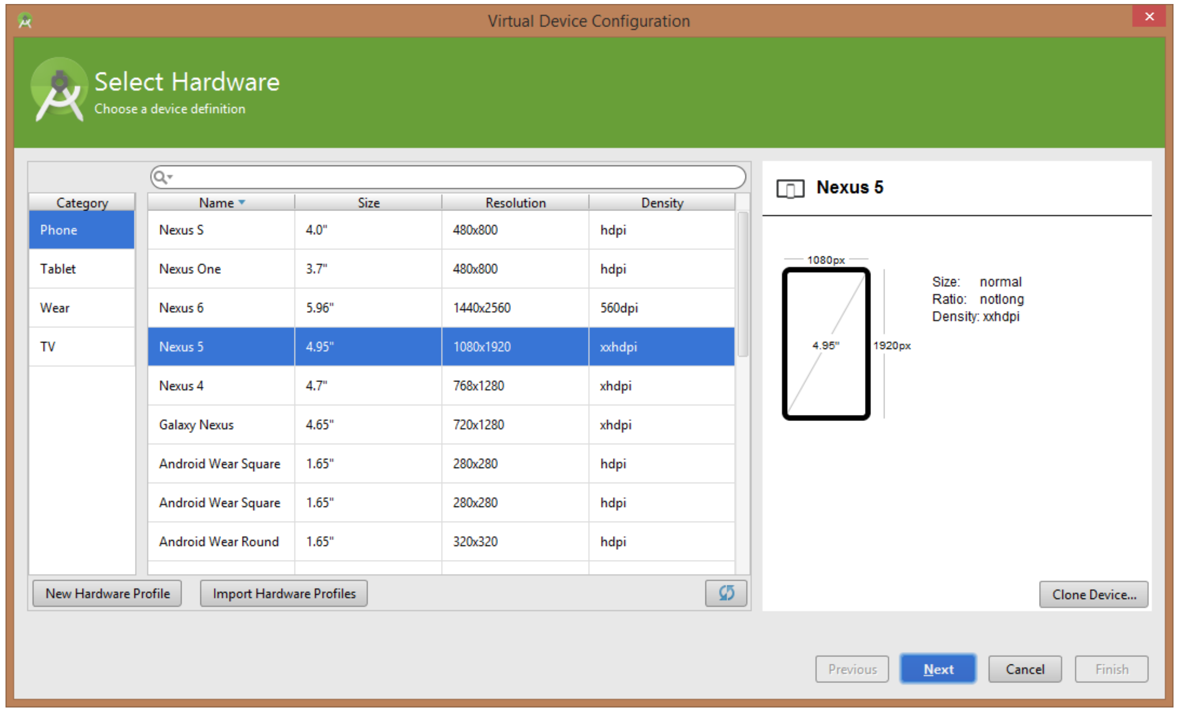
Your Virtual Devices
Android Studio

Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 5 API 21 x86	1080 × 1920: xxhdpi	21	Google APIs	x86	750 MB	
	TestAVD	240 × 320: ldpi	17	Android 4.2.2	arm	753 MB	

+ Create Virtual Device...

Create a New AVD

- Select from a list of known configurations
- Set AVD parameters



Virtual Device Configuration

Select Hardware

Choose a device definition

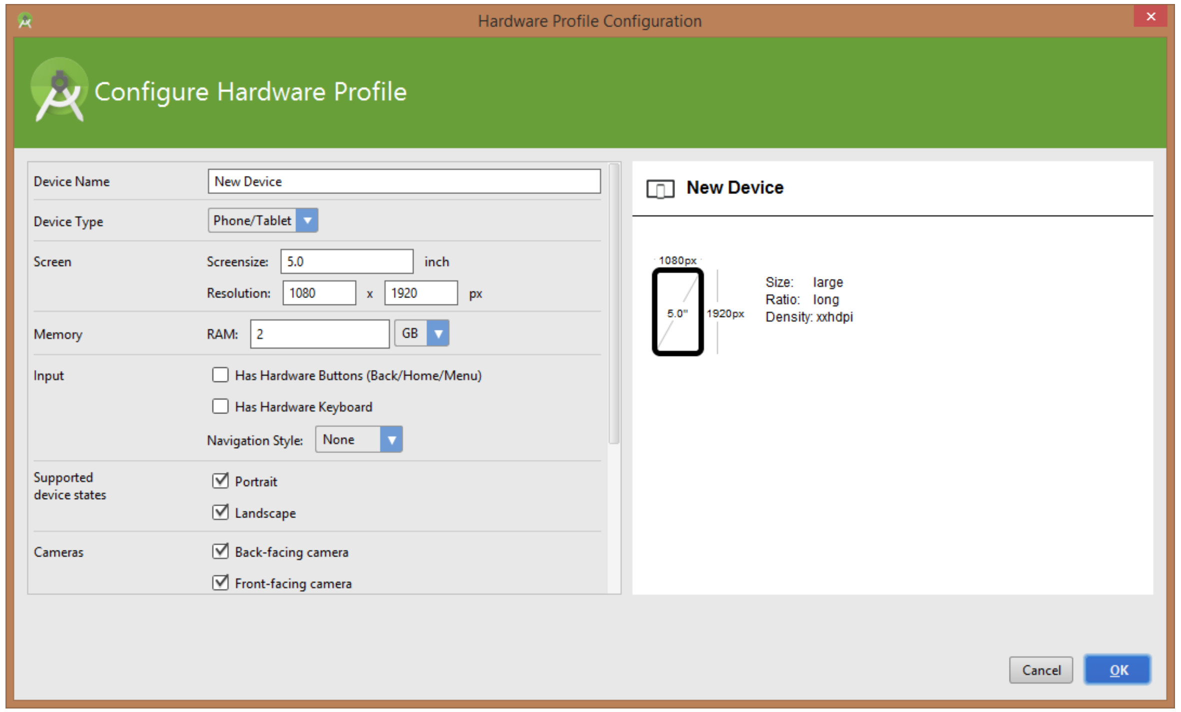
Category
Phone
Tablet
Wear
TV

Name	Size	Resolution	Density
Nexus S	4.0"	480x800	hdpi
Nexus One	3.7"	480x800	hdpi
Nexus 6	5.96"	1440x2560	560dpi
Nexus 5	4.95"	1080x1920	xxhdpi
Nexus 4	4.7"	768x1280	xhdpi
Galaxy Nexus	4.65"	720x1280	xhdpi
Android Wear Square	1.65"	280x280	hdpi
Android Wear Square	1.65"	280x280	hdpi
Android Wear Round	1.65"	320x320	hdpi

Nexus 5

1080px
4.95"
1920px

Size: normal
Ratio: notlong
Density: xxhdpi



Hardware Profile Configuration

Configure Hardware Profile

Device Name	New Device
Device Type	Phone/Tablet
Screen	Screensize: 5.0 inch
	Resolution: 1080 x 1920 px
Memory	RAM: 2 GB
Input	☐ Has Hardware Buttons (Back/Home/Menu)
	☐ Has Hardware Keyboard
	Navigation Style: None
Supported device states	☑ Portrait
	☑ Landscape
Cameras	☑ Back-facing camera
	☑ Front-facing camera

New Device

1080px
5.0"
1920px

Size: large
Ratio: long
Density: xxhdpi

Genymotion

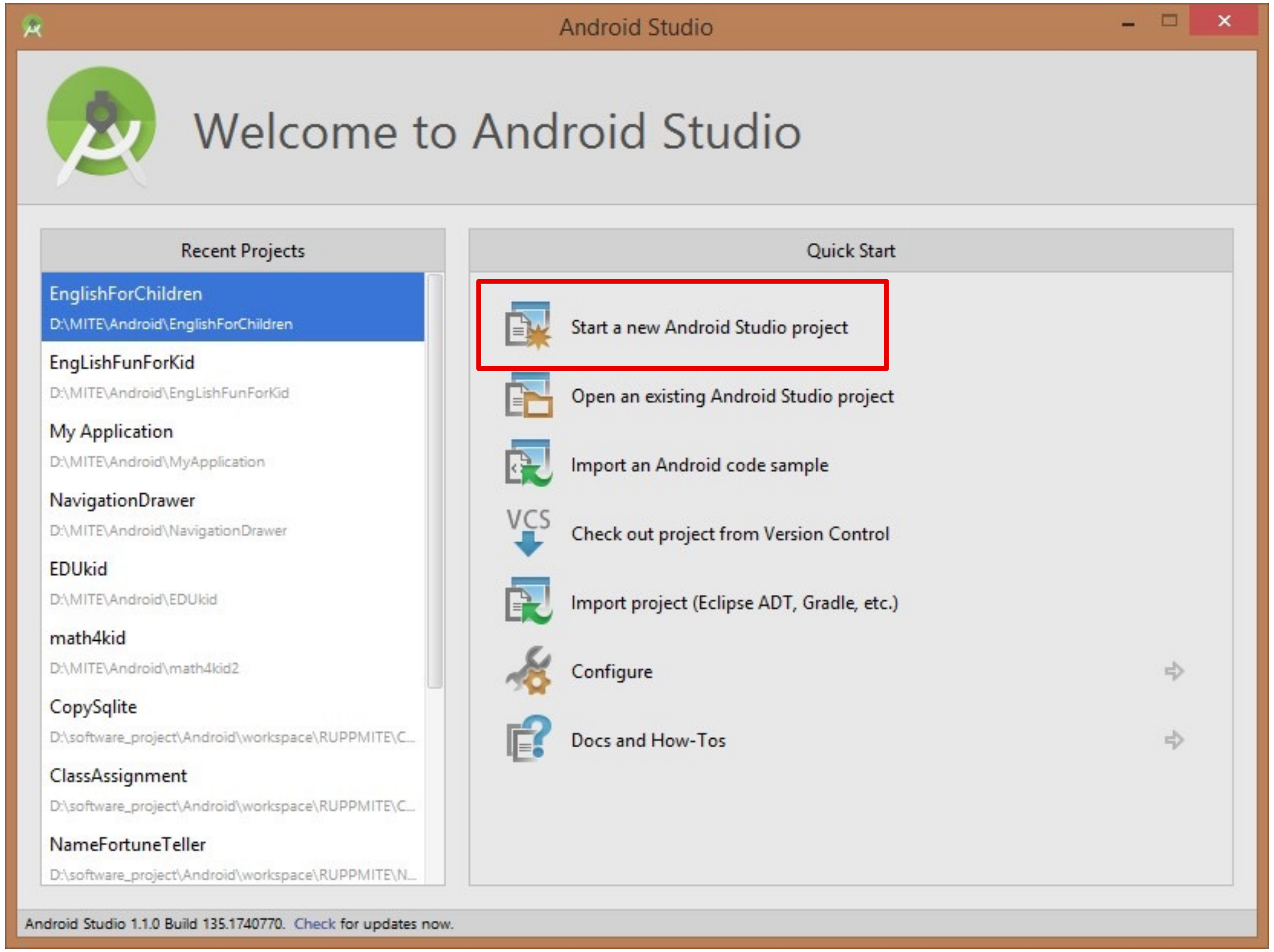
To install Genymotion plugin for Android Studio:

1. In Android Studio, go to **File** > **Settings**.
2. Select **Plugins** and click **Browse Repositories**.
3. Right-click on **Genymotion** and click **Download and install**.
4. To see Genymotion plugin icon, display the toolbar by clicking **View** > **Toolbar**.
5. To use this plugin, Genymotion must be installed on your system.

Create New Project

Launch Android Studio

- Start a new Android Studio project



Android Studio

Welcome to Android Studio

Recent Projects

EnglishForChildren
D:\MITE\Android\EnglishForChildren

EngLishFunForKid
D:\MITE\Android\EngLishFunForKid

My Application
D:\MITE\Android\MyApplication

NavigationDrawer
D:\MITE\Android\NavigationDrawer

EDUkid
D:\MITE\Android\EDUkid

math4kid
D:\MITE\Android\math4kid2

CopySqlite
D:\software_project\Android\workspace\RUPPMITE\C...

ClassAssignment
D:\software_project\Android\workspace\RUPPMITE\C...

NameFortuneTeller
D:\software_project\Android\workspace\RUPPMITE\N...

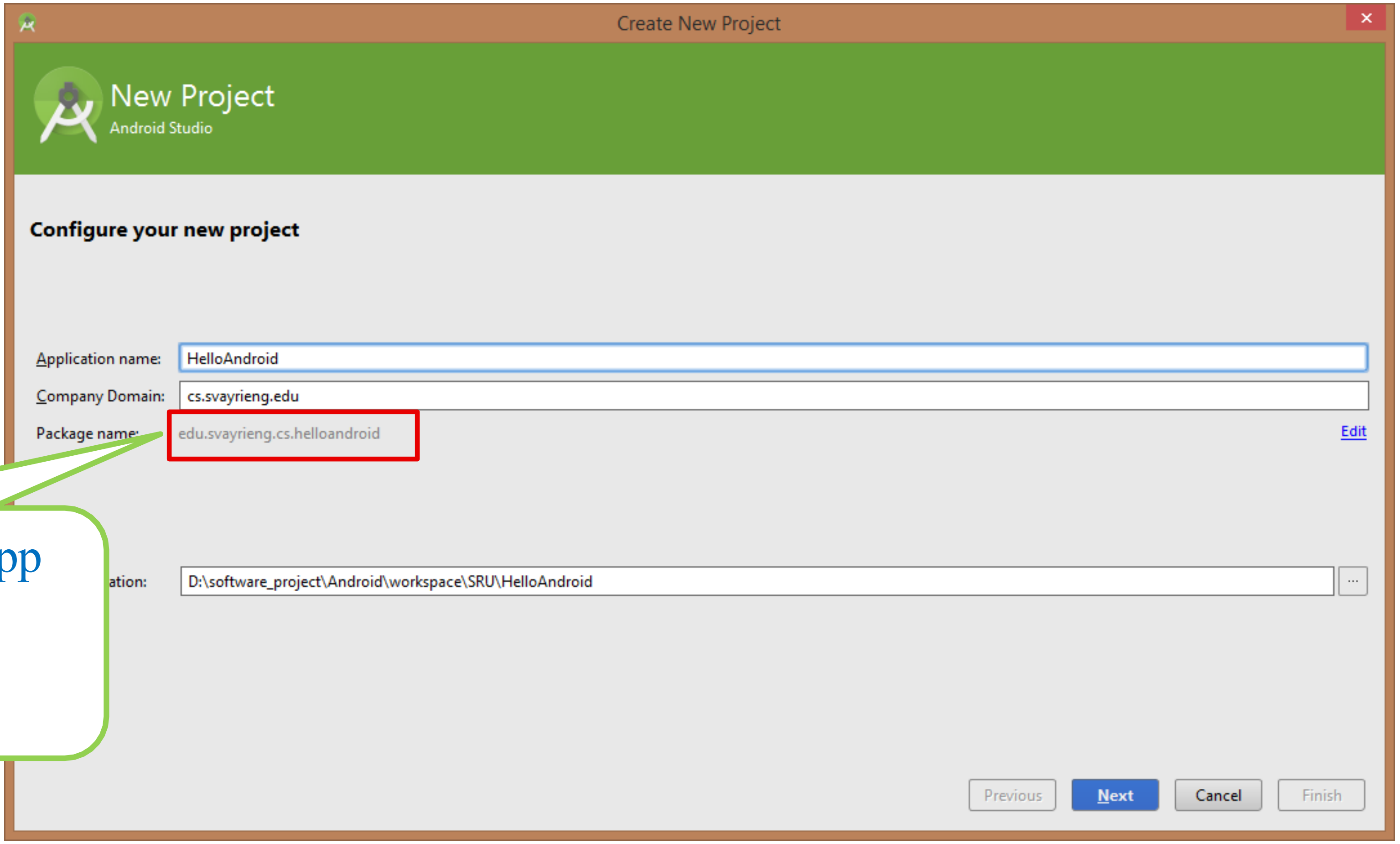
Quick Start

Start a new Android Studio project
Open an existing Android Studio project
Import an Android code sample
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Configure
Docs and How-Tos

Android Studio 1.1.0 Build 135.1740770. Check for updates now.

New Project – Configuration

- Choose
  - Application name
    - e.g., Hello Android
  - Company domain
    - e.g., cs.wu.edu
- Click *Next*



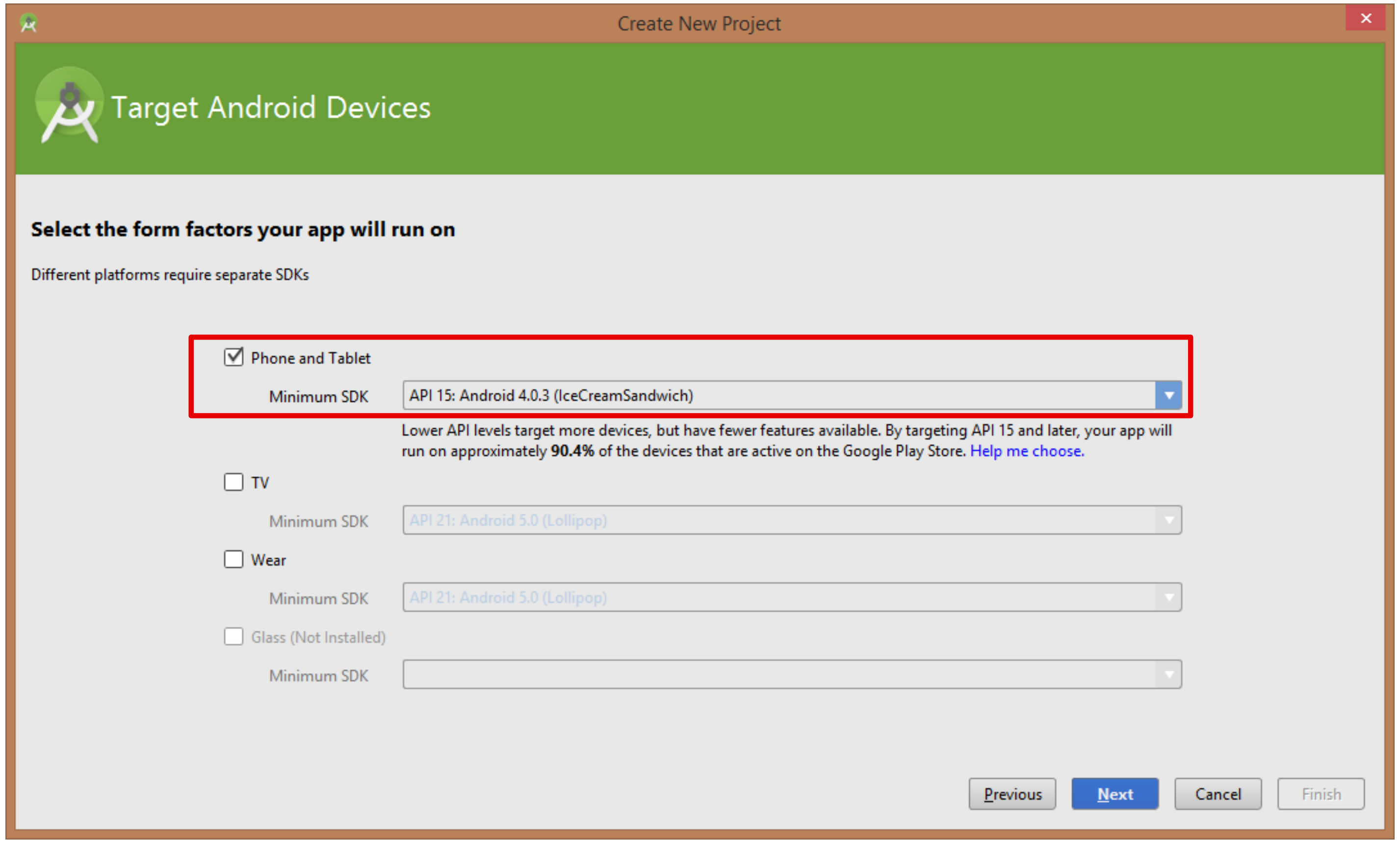
Configure your new project

Application name: HelloAndroid
Company Domain: cs.svayrieng.edu
Package name: edu.svayrieng.cs.helloandroid
ation: D:\software_project\Android\workspace\SRU\HelloAndroid

Package name is derived from app name and company domain. Must be unique across all installed apps.

New Project – Form Factor

- Choose the form factor
  - Phone and tablet
  - TV
  - Wear
  - Glass
- Choose the minimum SDK API level
- Click *Next*



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms 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 **90.4%** of the devices that are active on the Google Play Store. Help me choose.

☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Wear

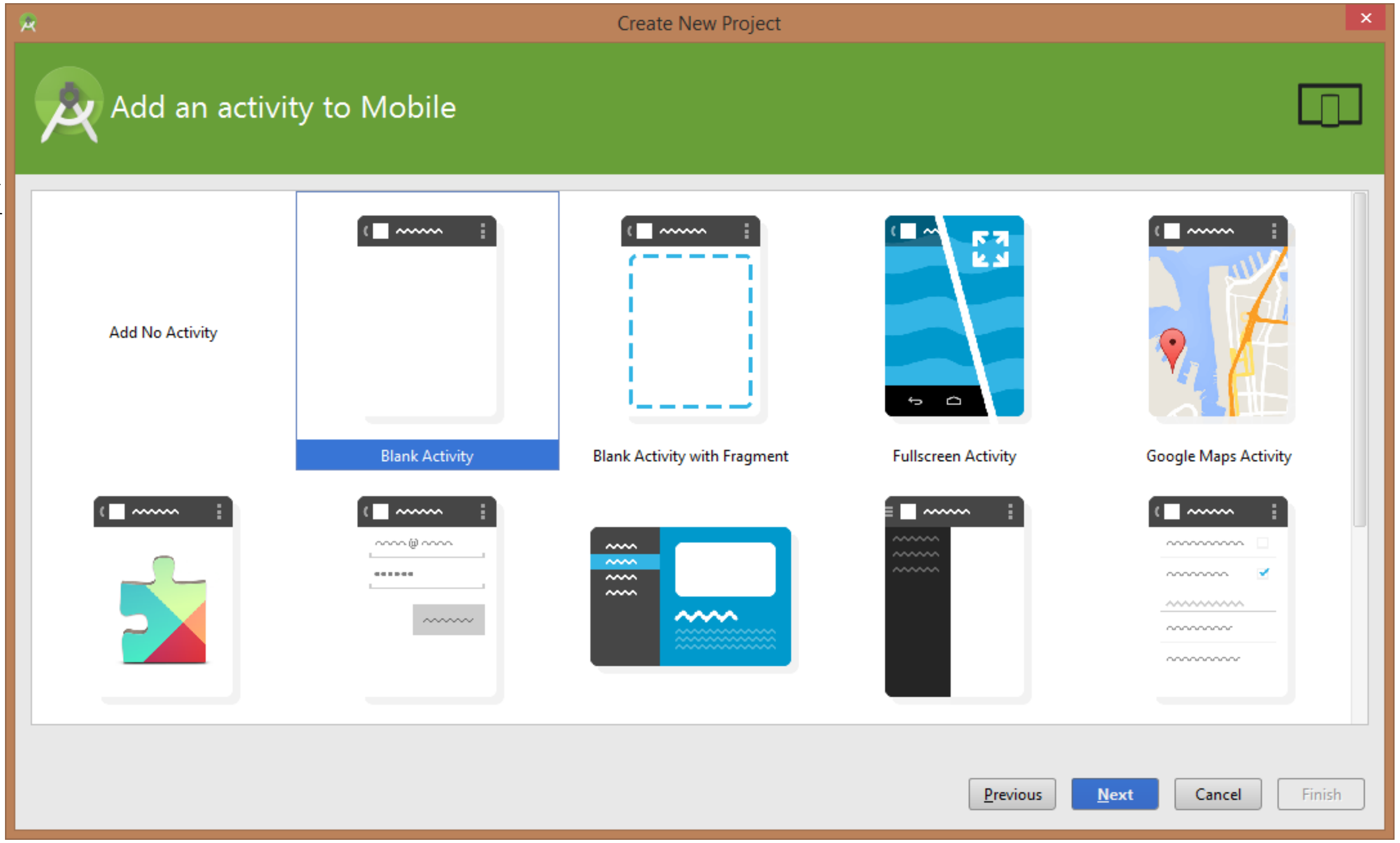
Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Glass (Not Installed)

Minimum SDK:

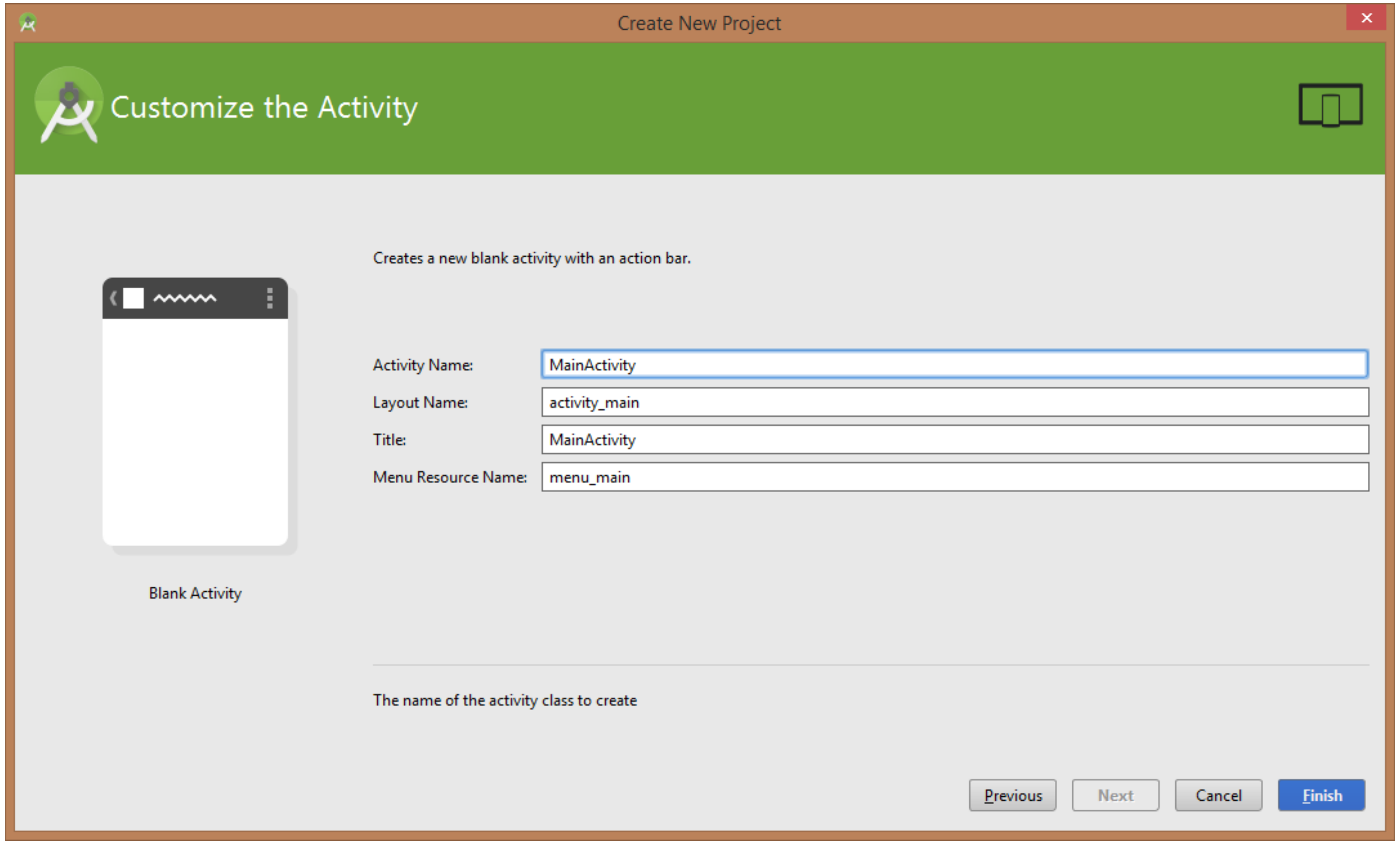
New Project – Activity Template

- Choose an Activity Template
- We will start with a Blank Activity
- Click Next



New Project – Blank Activity with an Action Bar

- Choose
  - Activity Name
  - Layout Name
  - Title
- Use the default for this app
- Click Finish



Create New Project

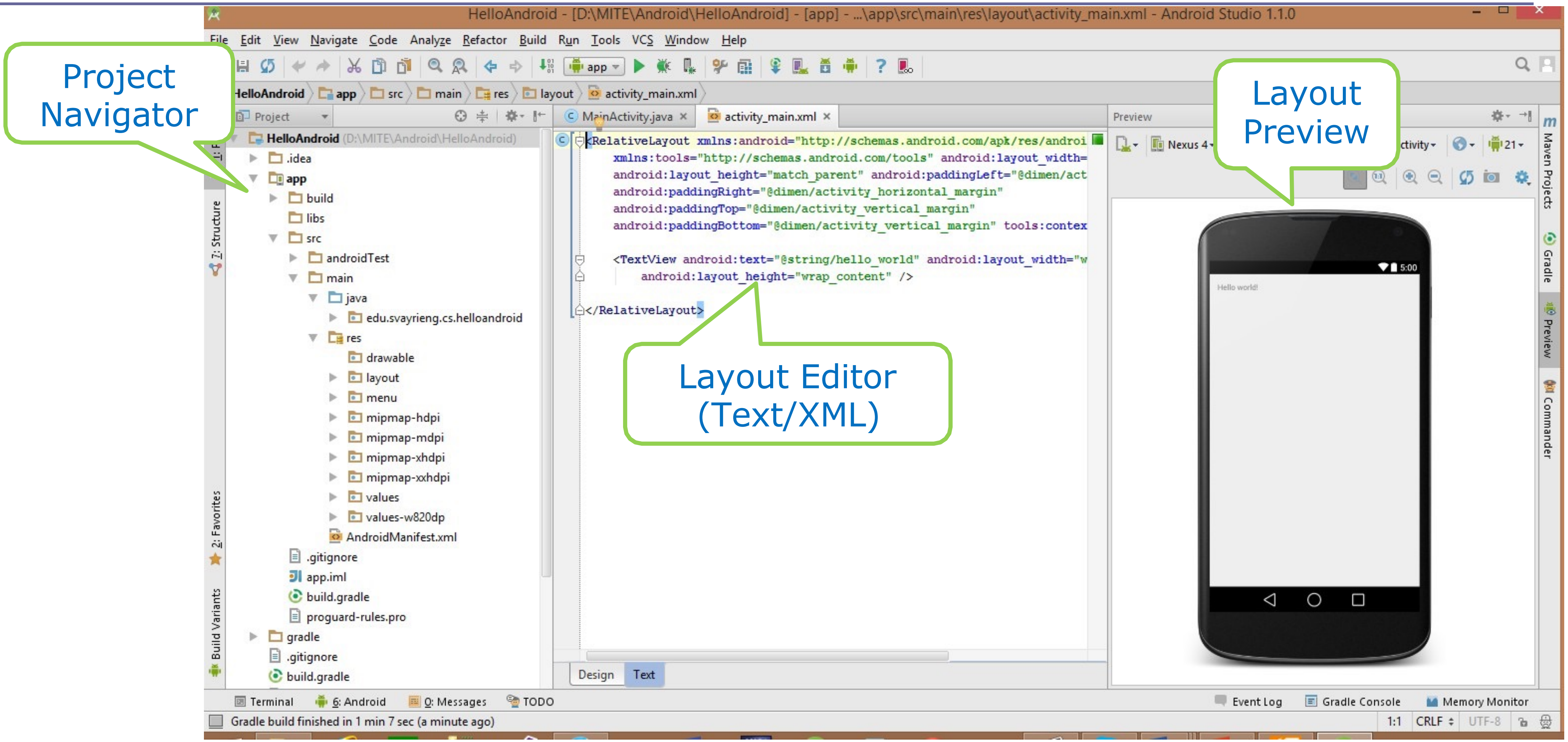
Customize the Activity

Creates a new blank activity with an action bar.

Activity Name: MainActivity
Layout Name: activity_main
Title: MainActivity
Menu Resource Name: menu_main

The name of the activity class to create

Android Project – Hello World!



HelloAndroid - [D:\MITE\Android\HelloAndroid] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 1.1.0

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HelloAndroid > app > src > main > res > layout > activity_main.xml

```
HelloAndroid (D:\MITE\Android\HelloAndroid)
  .idea
  app
    build
    libs
    src
      androidTest
      main
        java
          edu.svayrieng.cs.helloandroid
        res
          drawable
          layout
          menu
          mipmap-hdpi
          mipmap-mdpi
          mipmap-xhdpi
          mipmap-xxhdpi
          values
          values-w820dp
        AndroidManifest.xml
    .gitignore
    app.iml
    build.gradle
    proguard-rules.pro
  gradle
  .gitignore
  build.gradle
```

MainActivity.java | activity_main.xml

```xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/androi
    xmlns:tools="http://schemas.android.com/tools" android:layout_width=
    android:layout_height="match_parent" android:paddingLeft="@dimen/act
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:contex

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

</RelativeLayout>
```

Design | Text

Preview: Nexus 4 — Hello world!

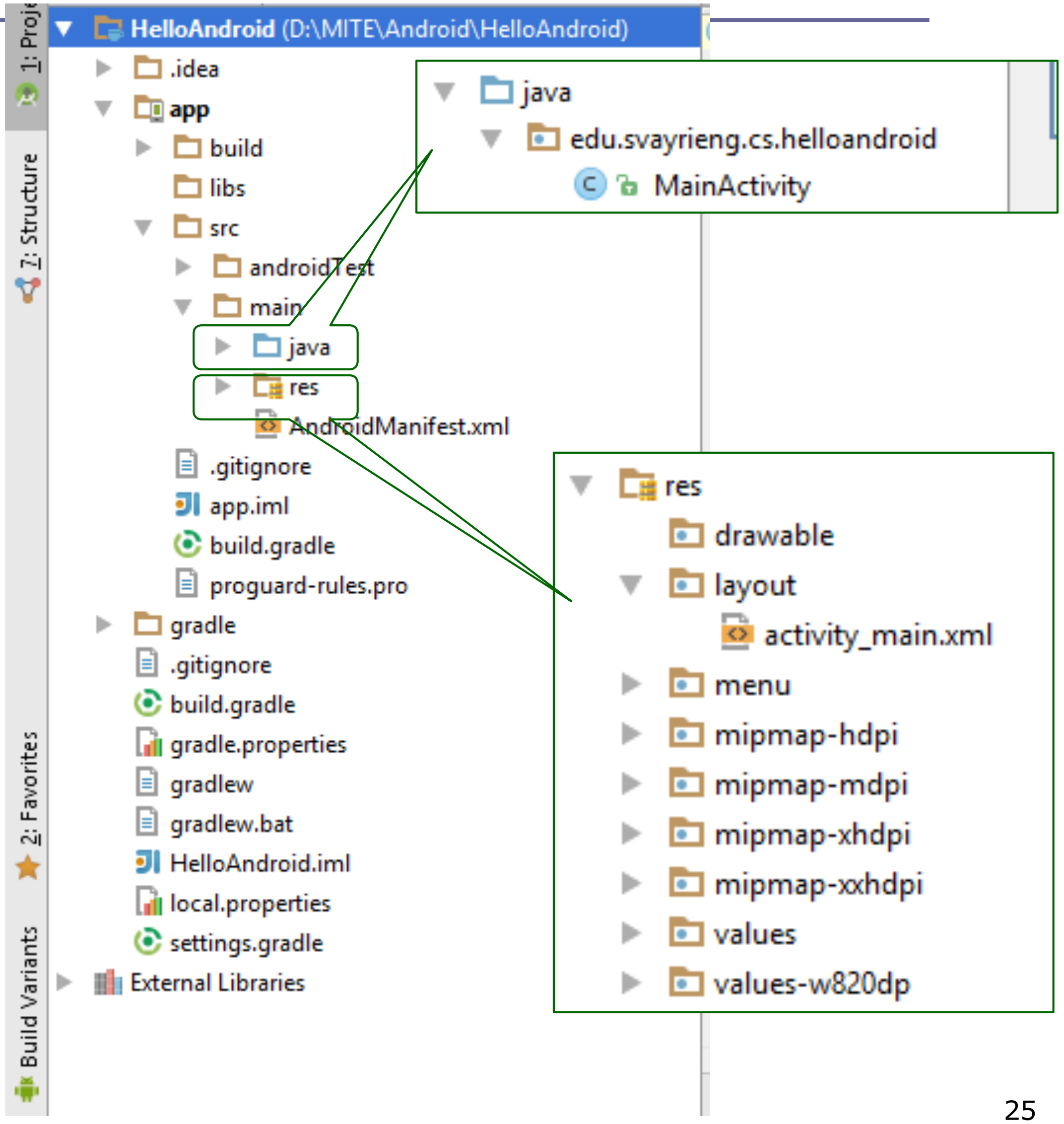
Terminal | 6: Android | 0: Messages | TODO | Event Log | Gradle Console | Memory Monitor

Gradle build finished in 1 min 7 sec (a minute ago) — 1:1 CRLF UTF-8

Anatomy of an Android Project

Android Project Structure

- Many files and folders are generated
- Let's focus on **app/src/main** folder
  - The main source folder
  - Contains the files to be edited
- Key files:
  - AndroidManifest.xml
  - res/layout/activity_my.xml
  - java/edu.wu.cs.helloandroid/
  - MyActivity.java



Elements of a Simple Android App

A simple Android app consists of

- An *app manifest* (XML file)
  - app/src/main/AndroidManifest.xml
- *App resources* (XML files)
  - Under app/src/main/res
  - layout, values, menu, drawable, **mipmap-***, etc.
- One or more *activities*. Each activity consists of
  - A Java class (Java file, under app/src/main/java), and
  - (Usually) an associated layout resource file (XML file)

Android App Manifest

- Required for every app
  - Must be in app/src/main/, i.e., the root directory
  - Must be named AndroidManifest.xml
- Include the essential information about the app
  - The components of the app
  - The main activity of the app, i.e., the launch point
  - System requirements to run the app
  - Permissions required to run the app
    - Internet, location, call, etc.

Android Activities

- *Activities* are the one of the building blocks of apps
  - A unit of single, focused task that a user can do
- An activity
  - Serves as an entry point to app
    - An app may have multiple entry points
  - Associated with a single screen of UI
    - UI can be created from resources or in code
  - Handles the responses to UI events
  - Interacts with other activities (inside or outside the app)
  - Responds to lifecycle events
    - (One of the) Smallest units that can be created or destroyed

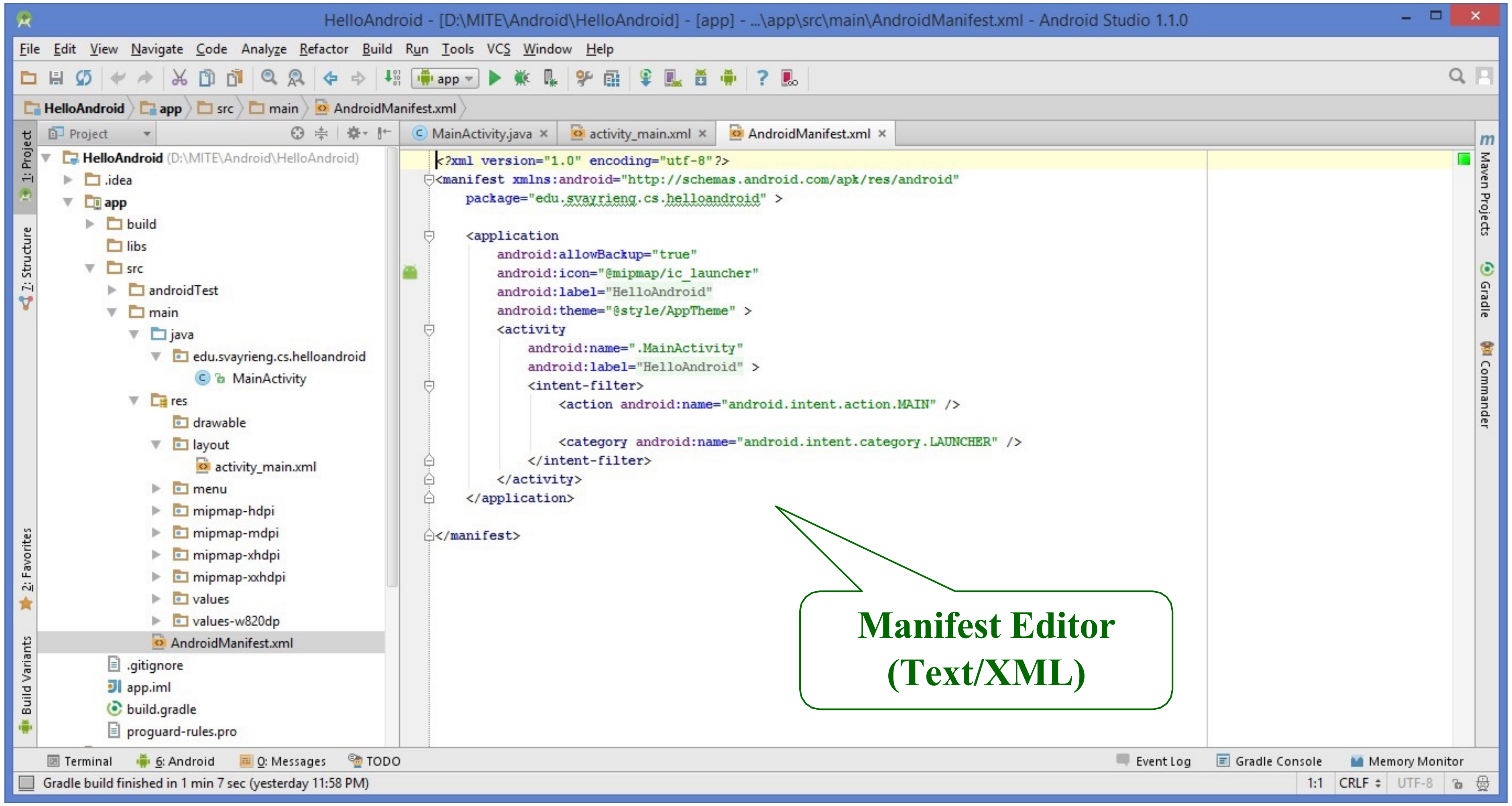
Android Resources

- Static data and contents used in the app
- Best practice: externalize resources from code
  - Accommodate different configurations, e.g., screen sizes
  - Support localization, different languages, e.g., French, Chinese
- Defined in XML files
  - In subfolders under res
  - Organized by types and configurations (default, alternatives)
- Resource types:
  - Layout, string, menu, drawable, animation, style, etc.

Android Resources – Layout

- Define the composition of UI in XML
  - The layout of a UI screen (of an activity), or
  - A component of the UI, which can be used elsewhere
- Stored in res/layout/
- A default layout file is created automatically
  - res/layout/activity_main.xml
  - It is associated with the main activity

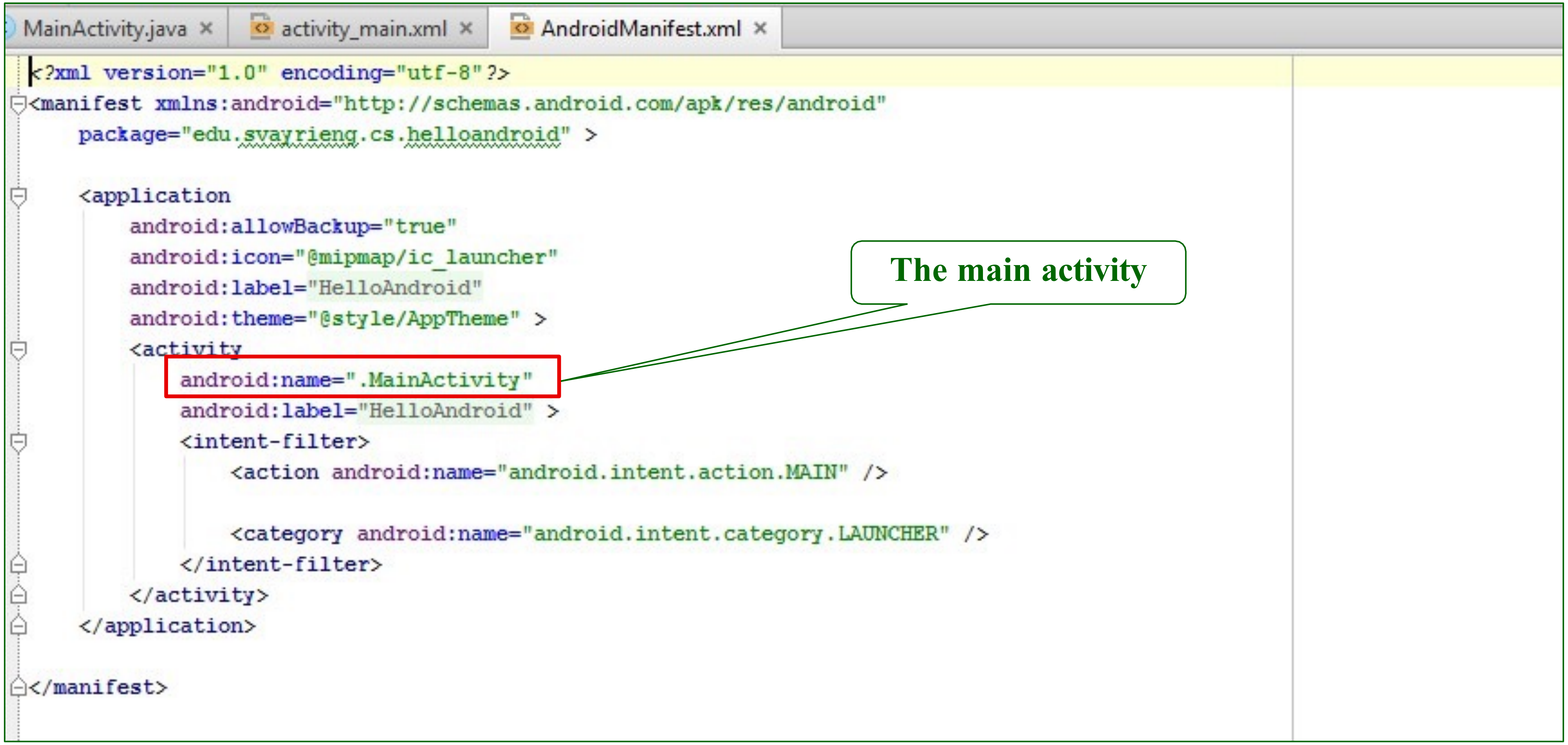
Hello Android – App Manifest



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.svayrieng.cs.helloandroid" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="HelloAndroid"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="HelloAndroid" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Gradle build finished in 1 min 7 sec (yesterday 11:58 PM)   1:1 CRLF UTF-8

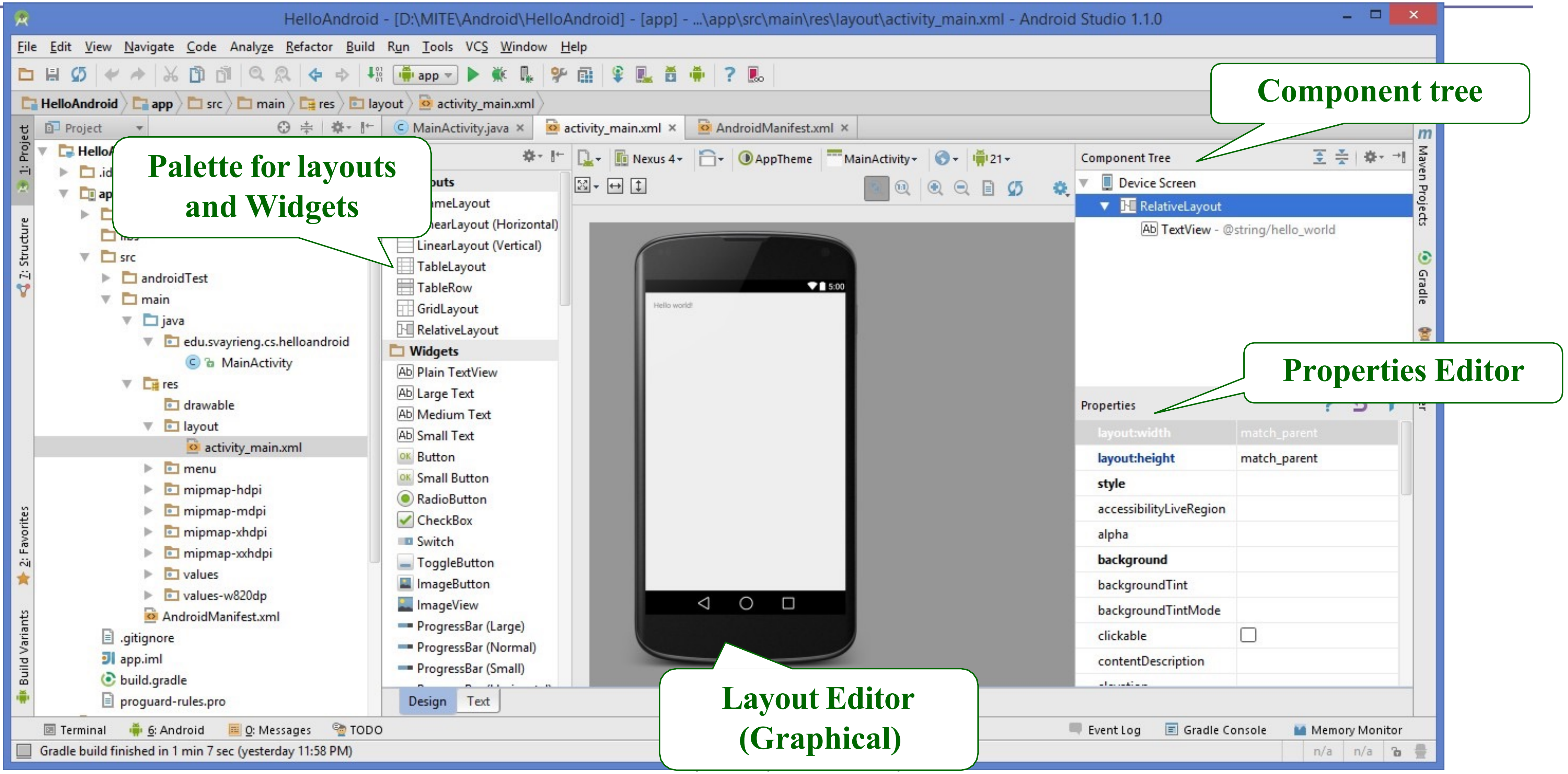
Hello Android – App Manifest



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.svayrieng.cs.helloandroid" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="HelloAndroid"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="HelloAndroid" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

The main activity

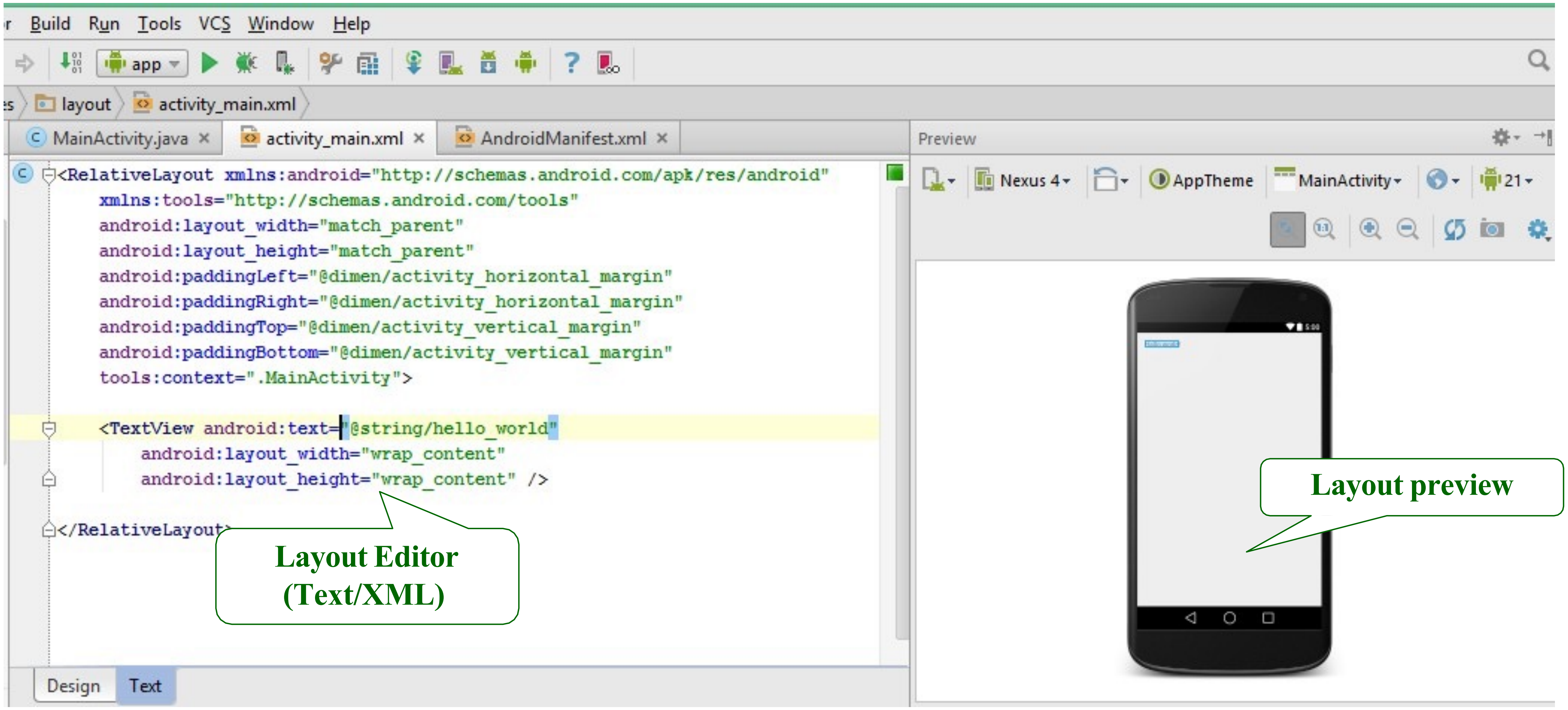
Hello Android – Layout The Graphical Design View


Component tree
Palette for layouts and Widgets
Properties Editor
Layout Editor (Graphical)
HelloAndroid - [D:\MITE\Android\HelloAndroid] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 1.1.0
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HelloAndroid app src main res layout activity_main.xml
MainActivity.java activity_main.xml AndroidManifest.xml
Nexus 4 AppTheme MainActivity 21
Component Tree
Device Screen
RelativeLayout
TextView - @string/hello_world
Properties
Design Text
Terminal 6: Android 0: Messages TODO Event Log Gradle Console Memory Monitor

layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveRegion	
alpha	
background	
backgroundTint	
backgroundTintMode	
clickable	☐
contentDescription	

Gradle build finished in 1 min 7 sec (yesterday 11:58 PM)

Hello Android – Layout
The Text Editor View

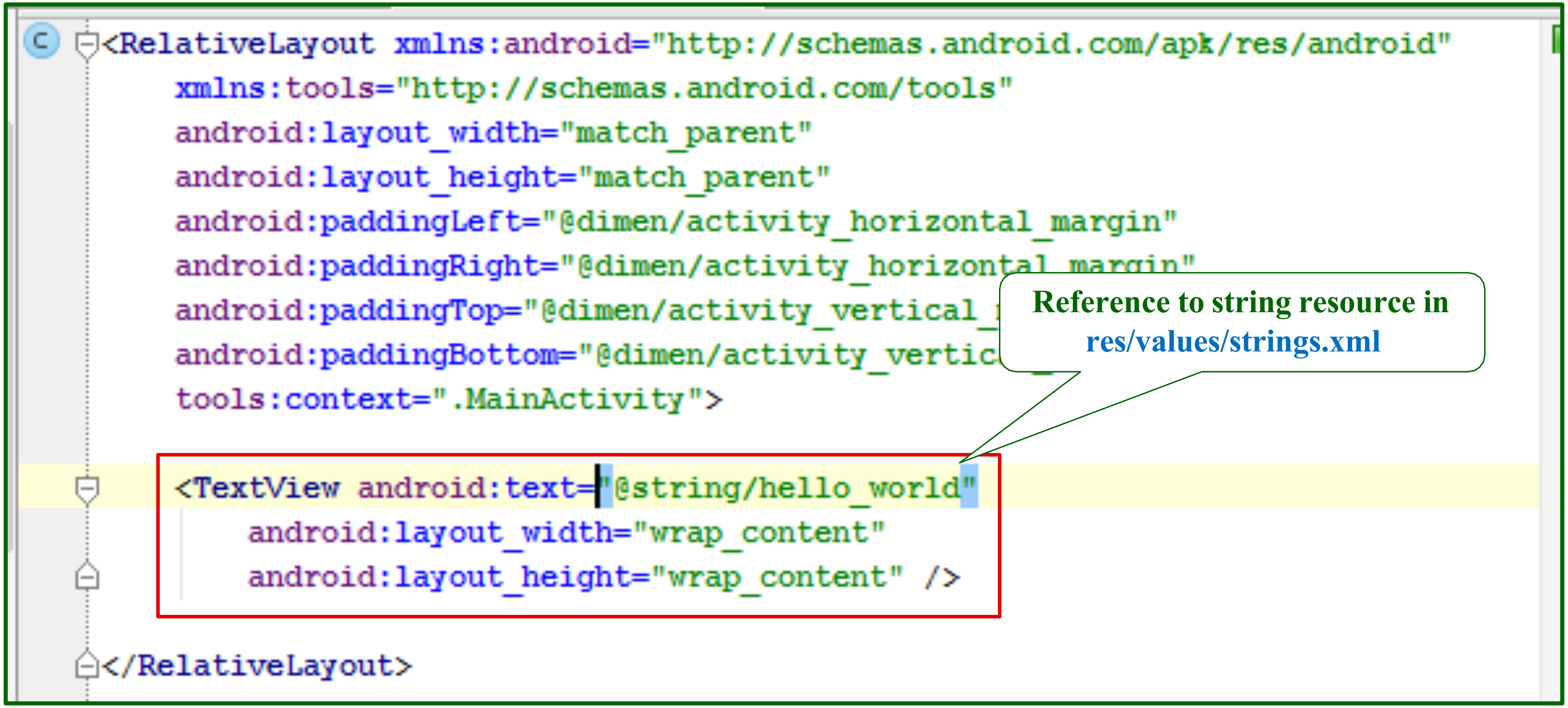


```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">

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

</RelativeLayout>
```

Layout preview

Layout Editor (Text/XML)

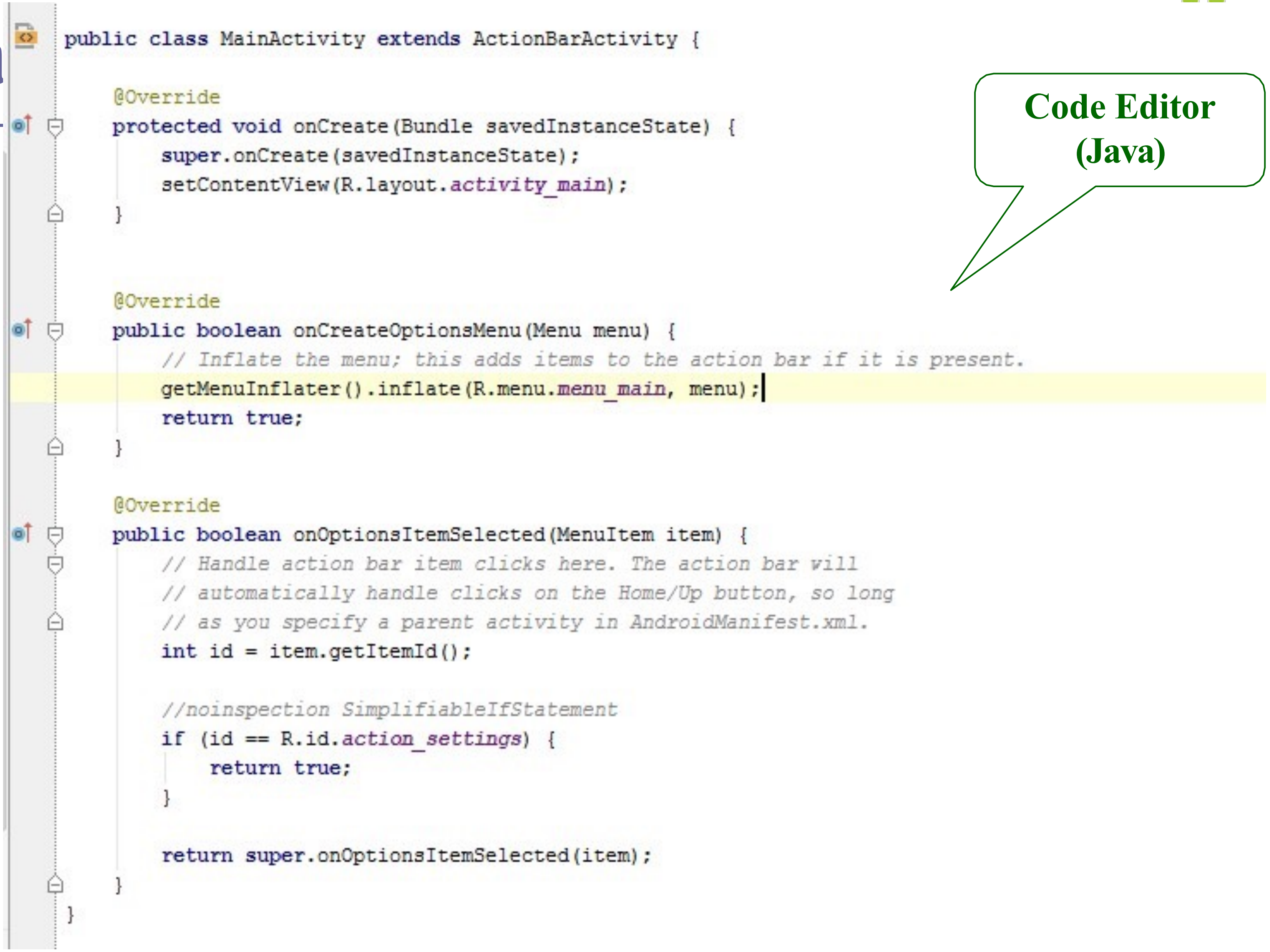
Hello Android – Layout activity_main.xml



```
<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_
    android:paddingBottom="@dimen/activity_vertic
    tools:context=".MainActivity">

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

</RelativeLayout>
```

Reference to string resource in res/values/strings.xml

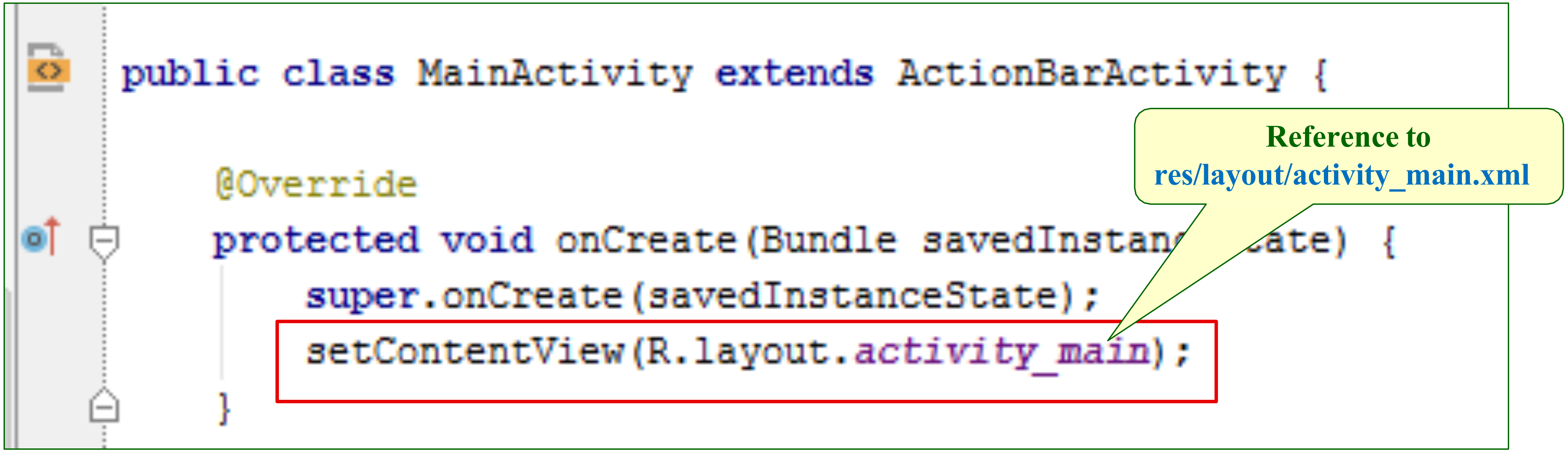
Hello Android – Activity MyActivity.java



```
public class MainActivity extends ActionBarActivity {

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


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

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

Code Editor (Java)

Hello Android – Activity MyActivity.java

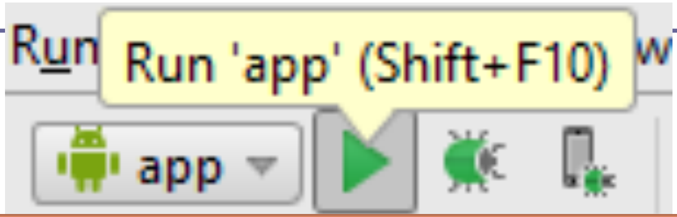


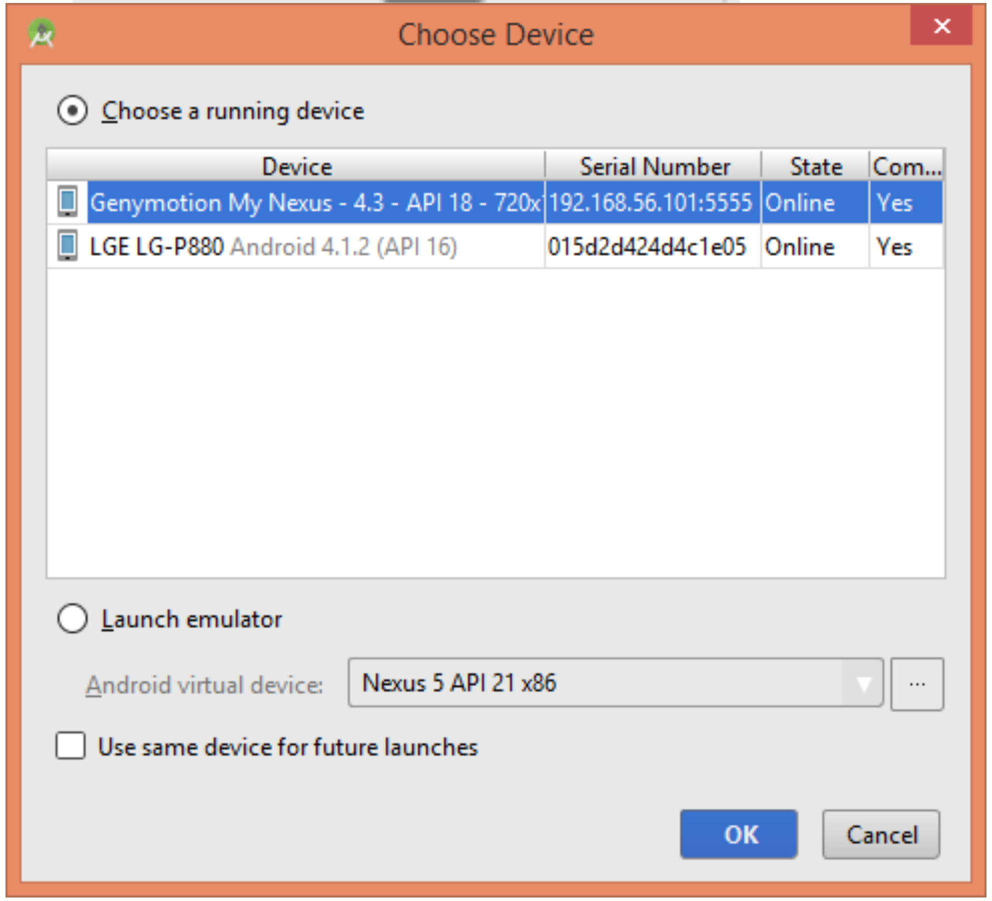
```
public class MainActivity extends ActionBarActivity {

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

Build and Run Android Apps

Run/Debug App on Emulator

- In Android Studio
  - Run | Run 'app', or
    - the *Run* icon in the toolbar
  - Run | Debug 'app', or
    - the *Debug* icon in the toolbar
- Select an AVD
  - Choose a running device, or
  - Launch an emulator




Choose Device

(●) Choose a running device

Device	Serial Number	State	Com...
Genymotion My Nexus - 4.3 - API 18 - 720x	192.168.56.101:5555	Online	Yes
LGE LG-P880 Android 4.1.2 (API 16)	015d2d424d4c1e05	Online	Yes

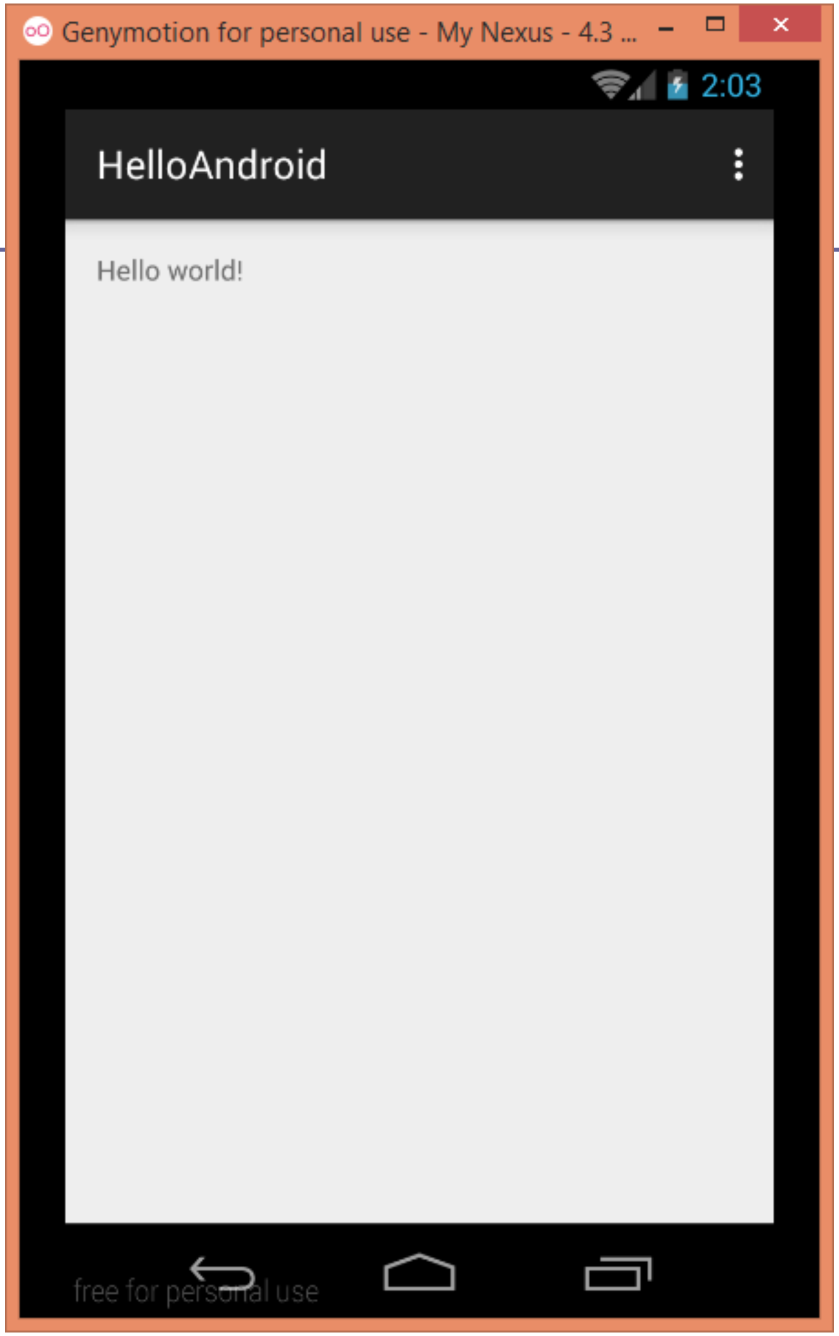
( ) Launch emulator

Android virtual device: Nexus 5 API 21 x86

☐ Use same device for future launches

OK Cancel

Run App on Emulator

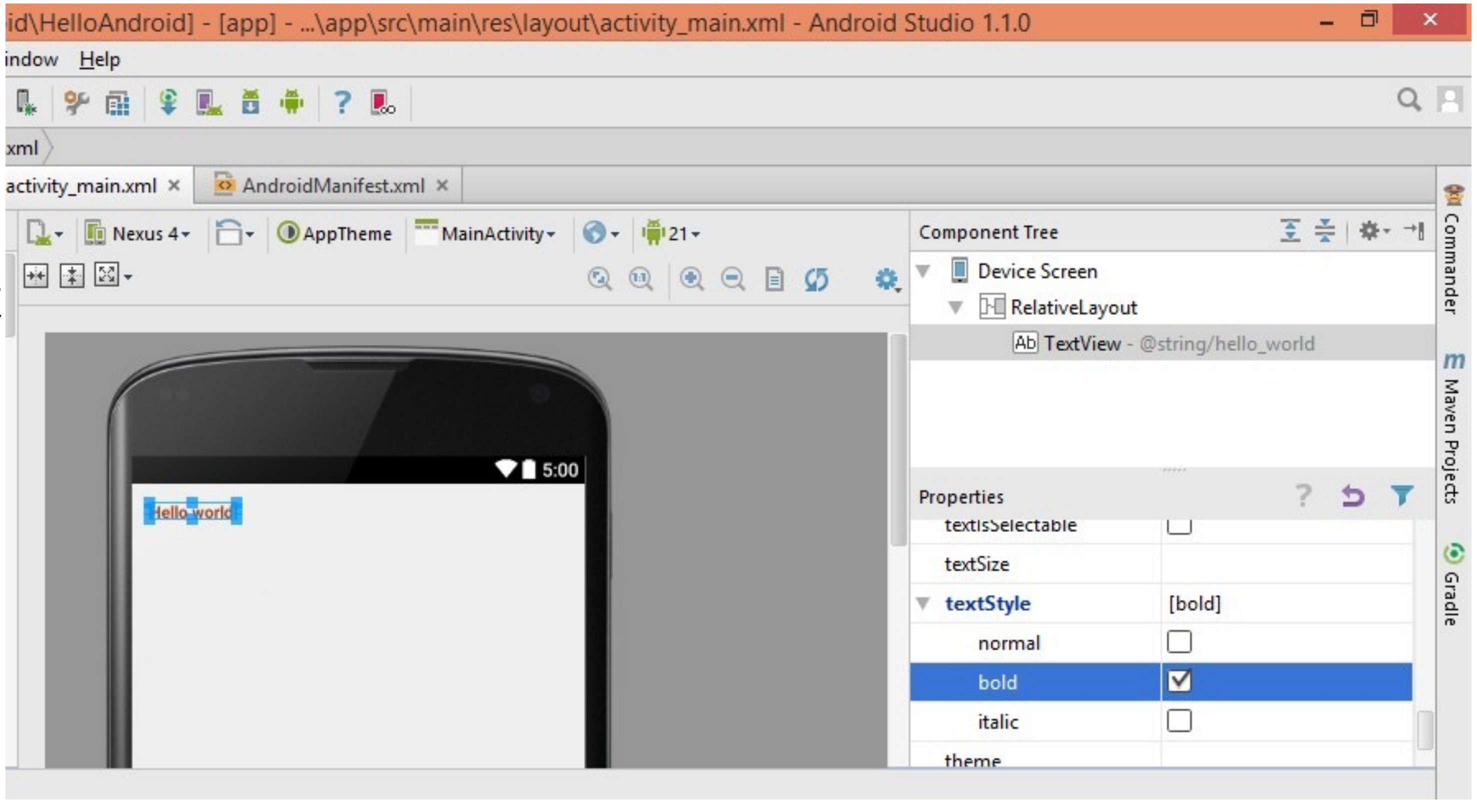


Hello world!

Edit App Resources

Edit Widget Properties

- Open **layout/activity_my.xml**
- Select the Designview
- Select the Text View widget
  - Text View displays a static text. Often known as a label.
- Edit the attributes
  - textColor
  - textSize
  - textStyle

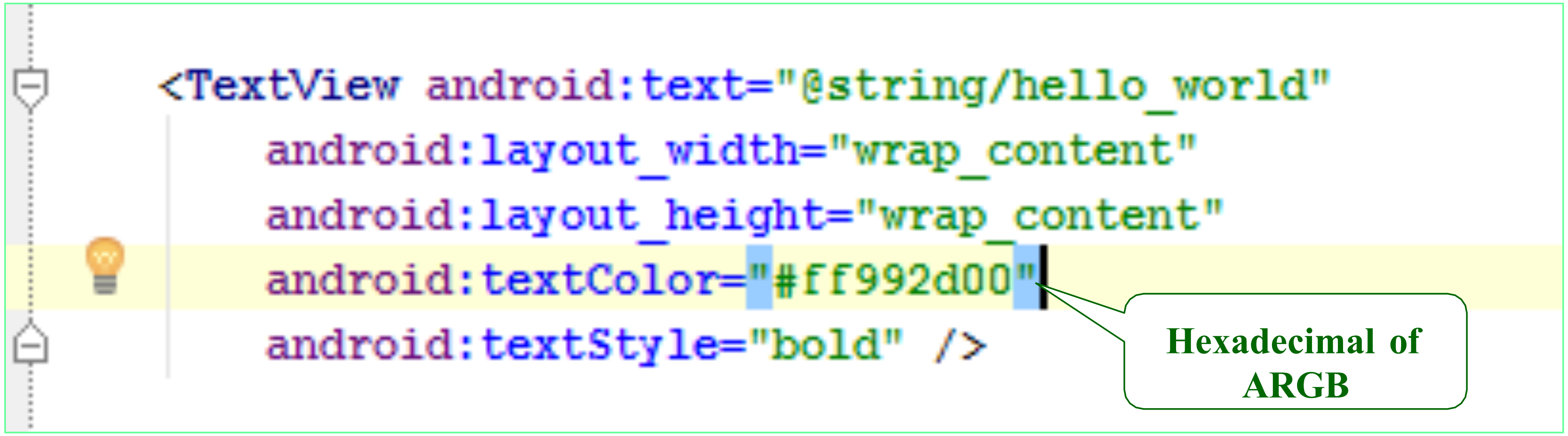

id\HelloAndroid] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 1.1.0

Component Tree
- Device Screen
  - RelativeLayout
    - TextView - @string/hello_world

Properties

textIsSelectable	☐
textSize	
textStyle	[bold]
normal	☐
bold	✓
italic	☐
theme	

The Text View – Change Text Styles

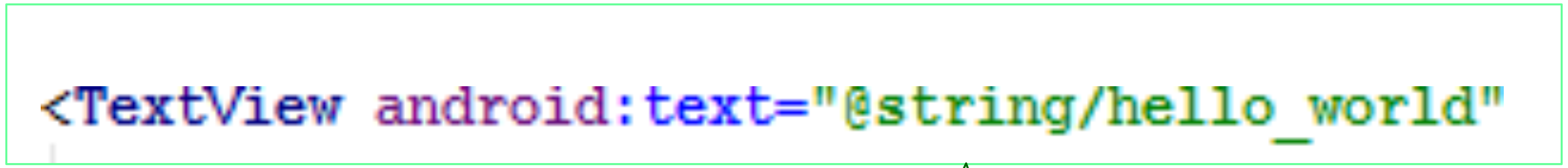


```
<TextView android:text="@string/hello_world"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#ff992d00"
    android:textStyle="bold" />
```

Hexadecimal of ARGB

Attributes of String Values

- The text attribute refers to a string resource



```
<TextView android:text="@string/hello_world"
```

Strings begin with @ are references to resources (defined in XML)

Homework

- Create the Android Project
- Edit resource to change the Hello World! To
  - Your name
  - Your Major
  - Academic Year
- And Runnable on your Emulator

Thank you!