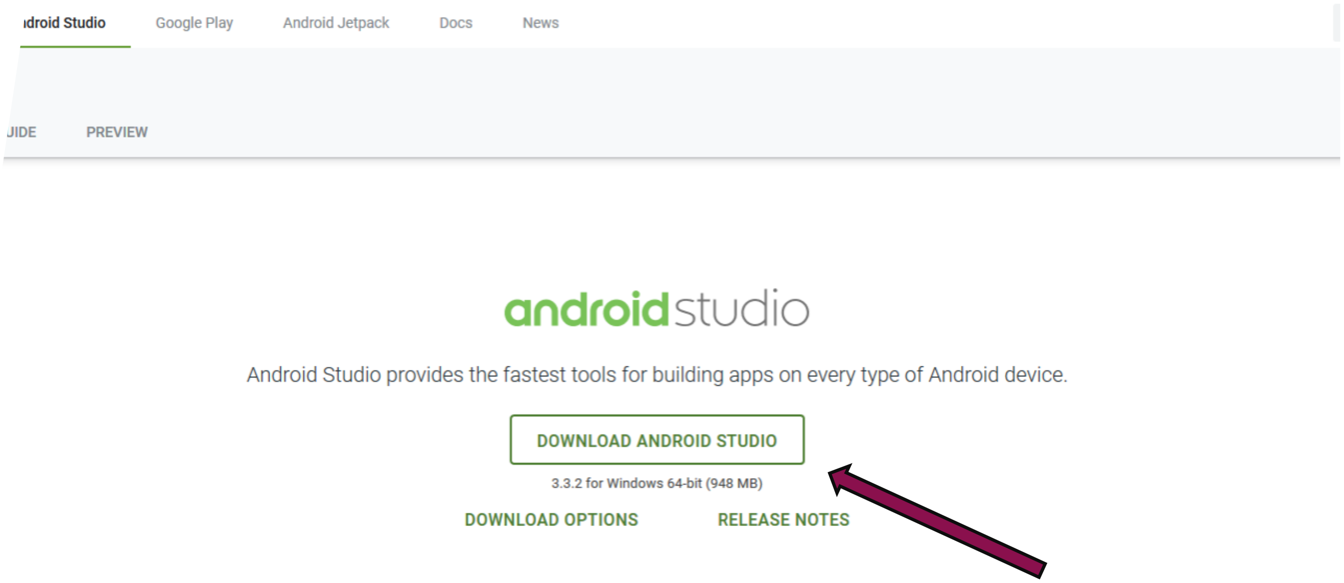
Android Application Development

Winter 2019



Android Studio provides the fastest tools for building apps on every type of Android device.

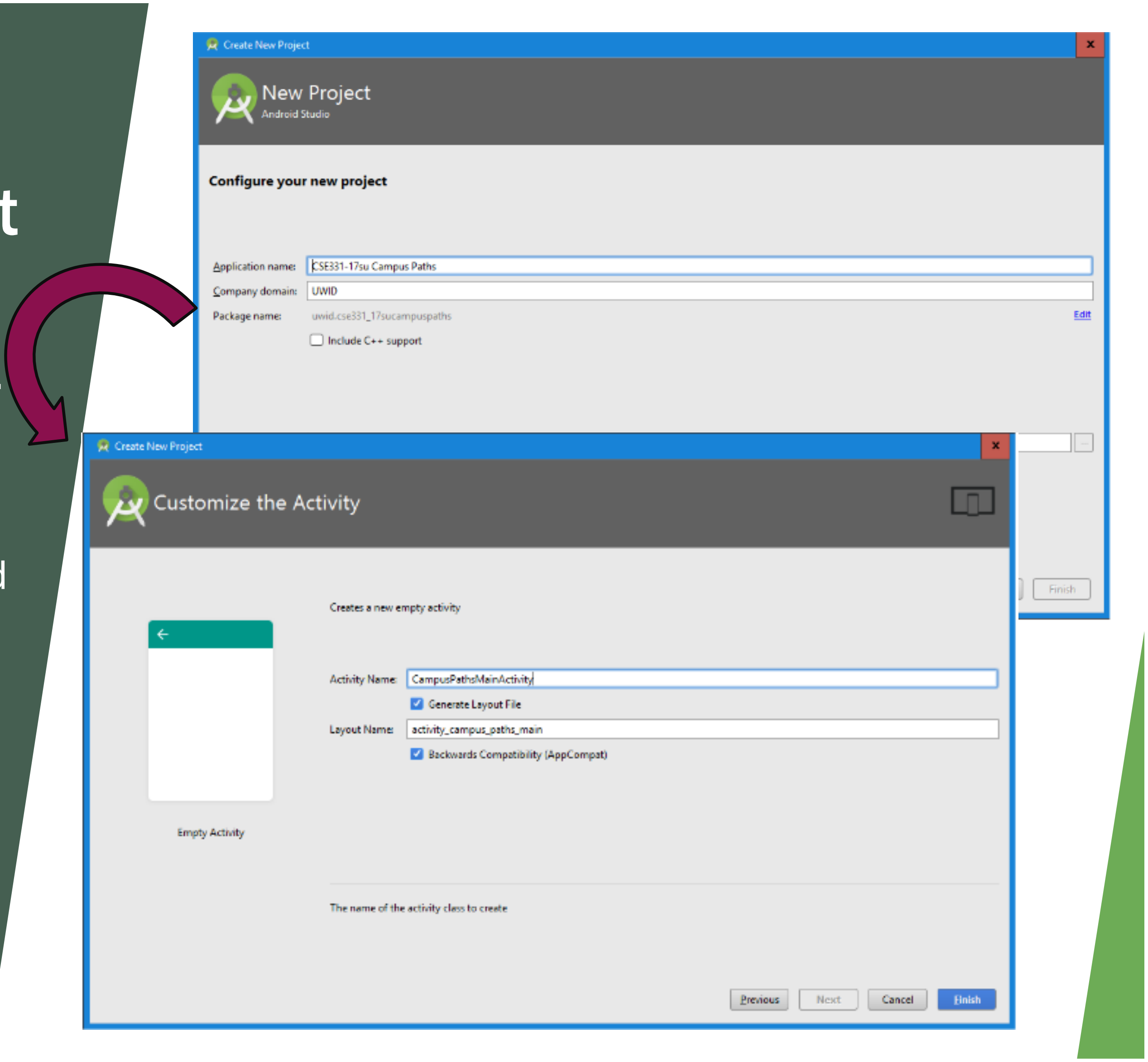
3.3.2 for Windows 64-bit (948 MB)

Install Android Studio

Navigate to: https://developer.android.com/studio/index.html

Create a New Project

- File → New → New Project…
- Choose your Minimum SDK
  - API 15: Android 4.0.3 (IceCreamSandwich) should work for Homework 9
- Create an Empty Activity



Configure your new project

Application name: CSE331-17su Campus Paths

Company domain: UWID

Package name: uwid.cse331_17sucampuspaths

□ Include C++ support

Creates a new empty activity

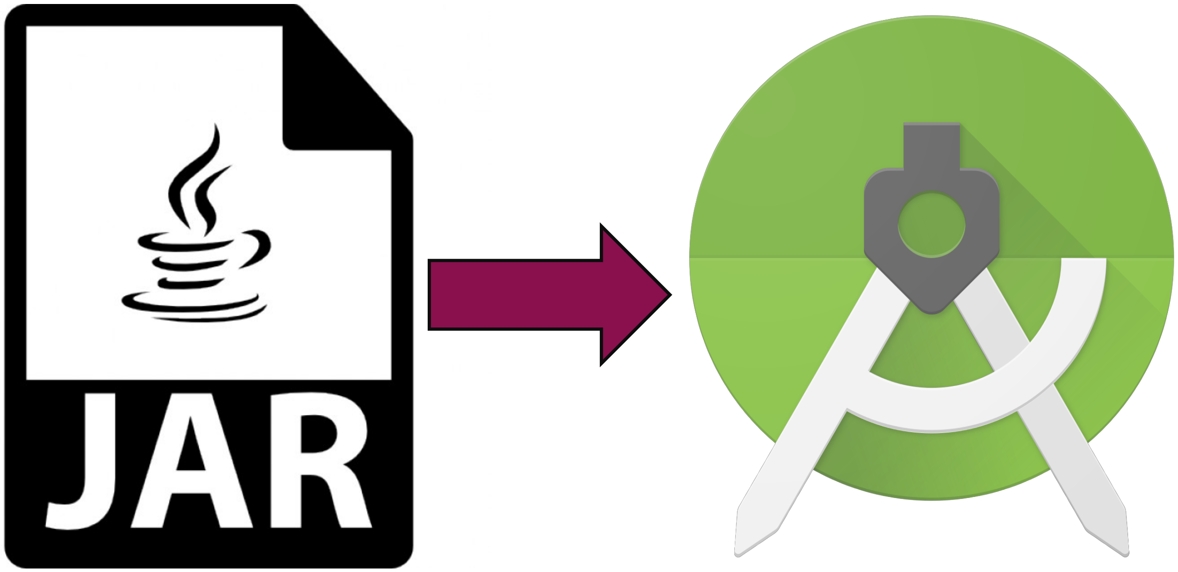
Activity Name: CampusPathsMainActivity

☑ Generate Layout File

Layout Name: activity_campus_paths_main

☑ Backwards Compatibility (AppCompat)

The name of the activity class to create

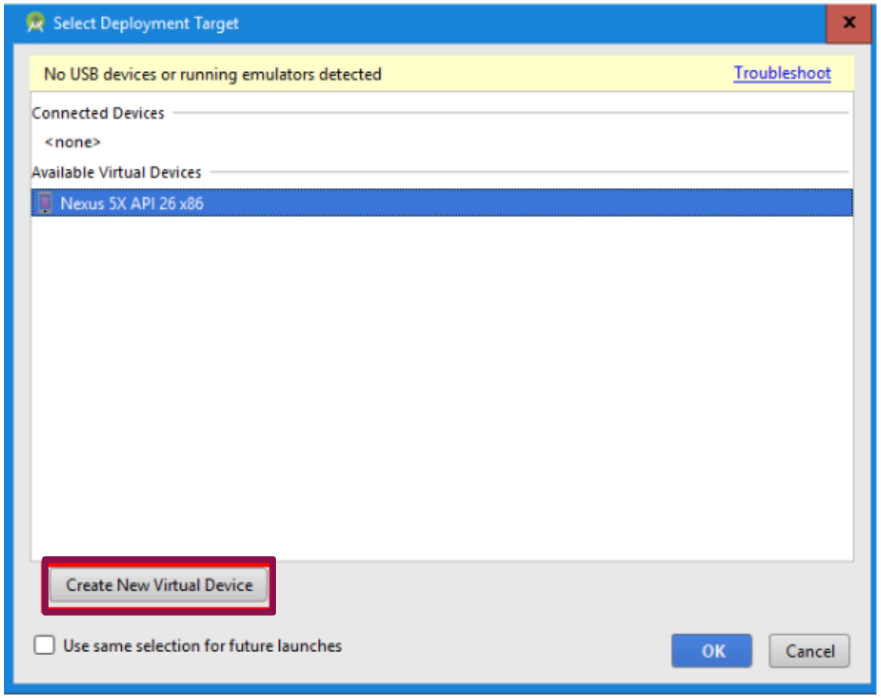


Import Previous Homework as a JAR

Step-by-step instructions available in Homework 9 specification!

Install Android Device Emulator

- Install via Android Studio
- For grading, we'll use the **Nexus 5X API 26** emulator
  - Make sure your application runs on this device!
- You can also connect your own Android devices to your computer to test your application!



Select Deployment Target

No USB devices or running emulators detected

Connected Devices

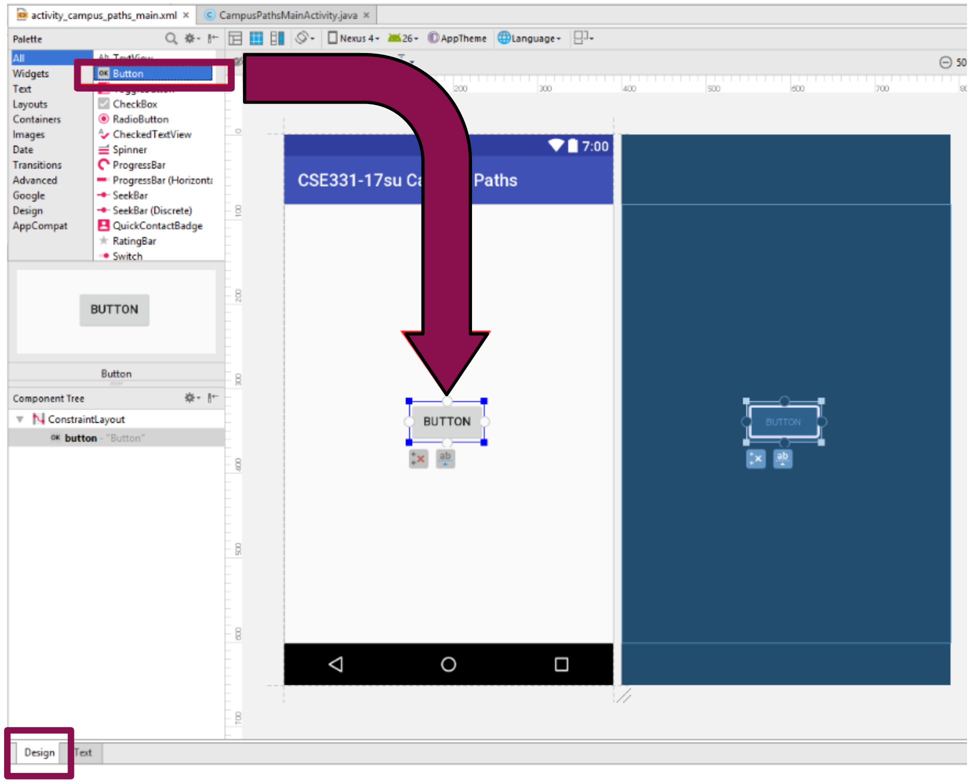
<none>

Available Virtual Devices

Nexus 5X API 26 x86

☐ Use same selection for future launches

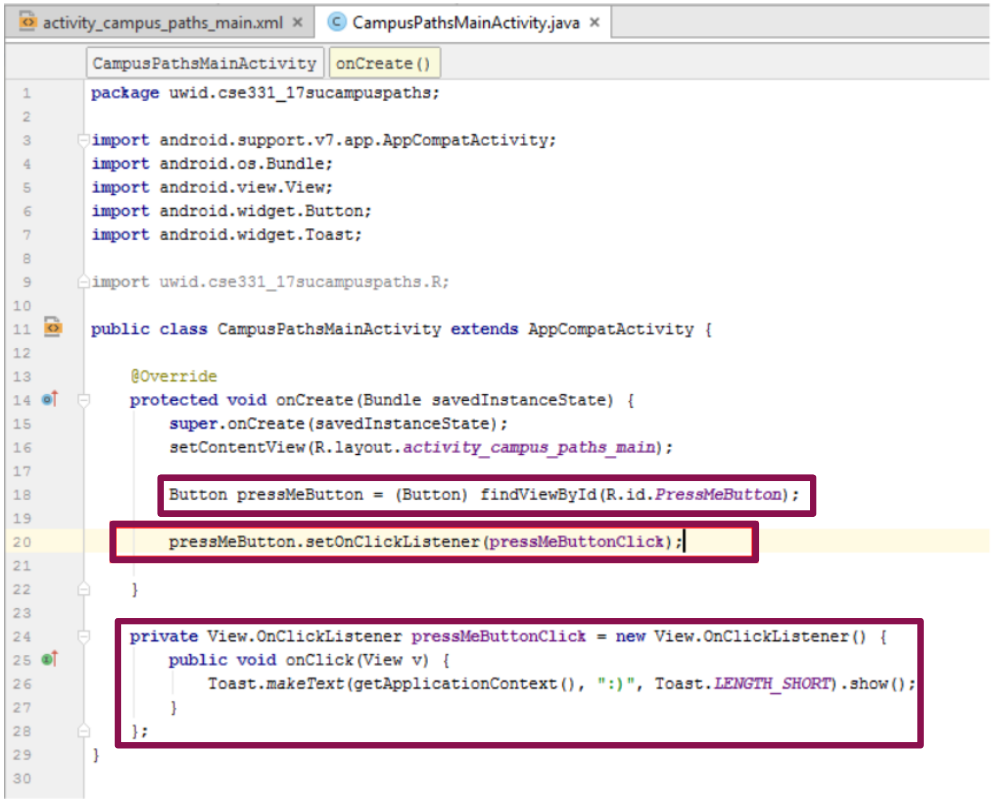
Add Widgets to Application



- Navigate to the "Design" tab
- Drag-and-drop any Android "widgets" observers to your layout
- This is where you build the view

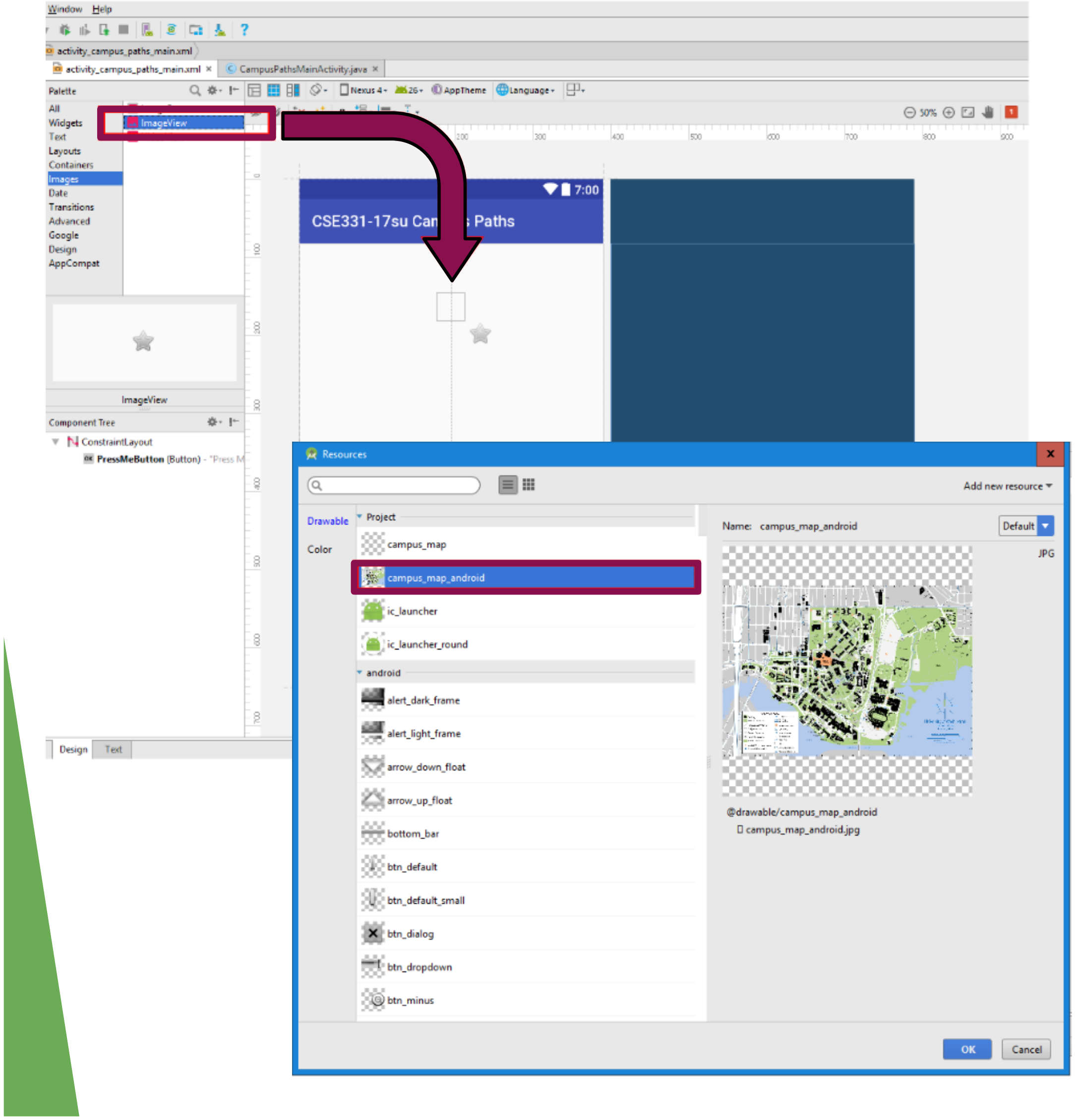
Attach Listeners to Widgets

- Lookup widgets by View ID in the “main” activity of your application
- Create a listener with code that will execute when event occurs
- Attach the listener to the widget



activity_campus_paths_main.xml ×
CampusPathsMainActivity.java ×
CampusPathsMainActivity
onCreate()

```java
package uwid.cse331_17sucampuspaths;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import uwid.cse331_17sucampuspaths.R;

public class CampusPathsMainActivity extends AppCompatActivity {

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

        Button pressMeButton = (Button) findViewById(R.id.PressMeButton);

        pressMeButton.setOnClickListener(pressMeButtonClick);

    }

    private View.OnClickListener pressMeButtonClick = new View.OnClickListener() {
        public void onClick(View v) {
            Toast.makeText(getApplicationContext(), ":)", Toast.LENGTH_SHORT).show();
        }
    };
}
```

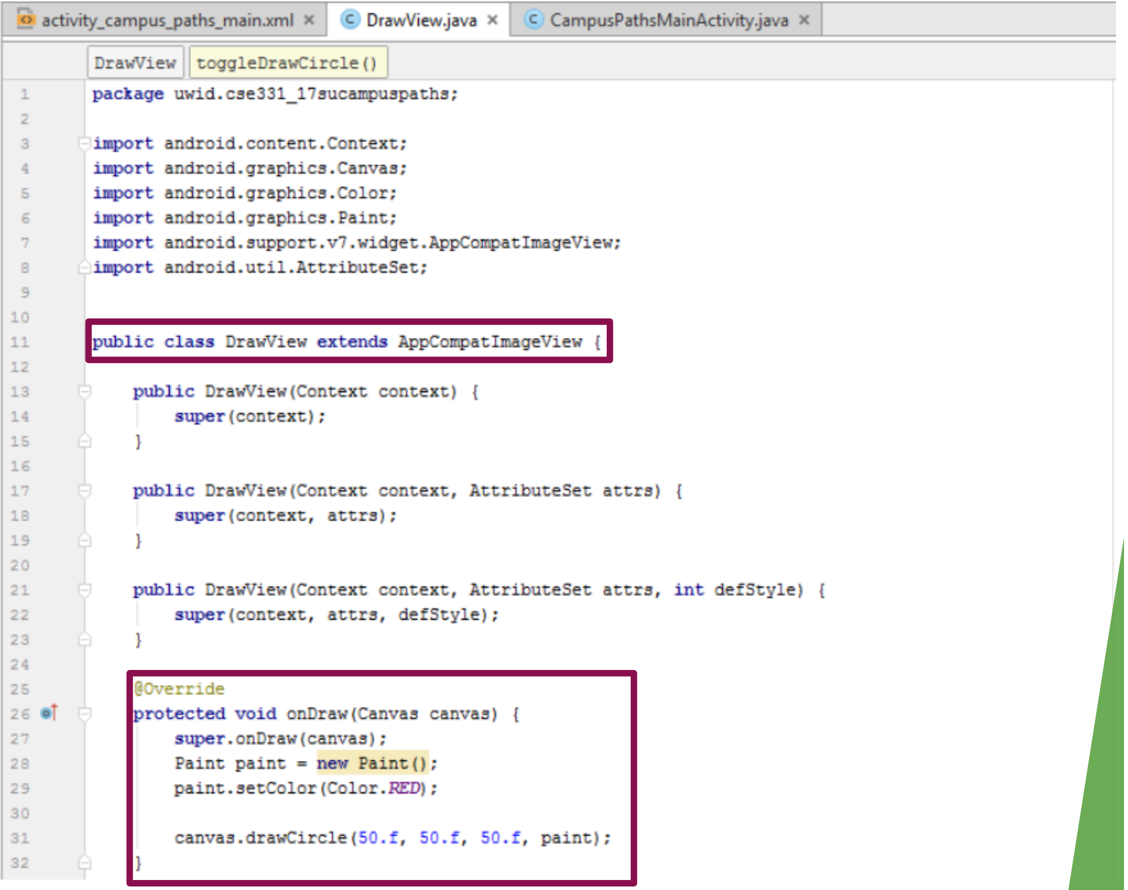


Add Images to Application

- Drag in an ImageView widget in the “Design” tab
- Select the campus map image designed for Android applications

Set up Image to be Drawn

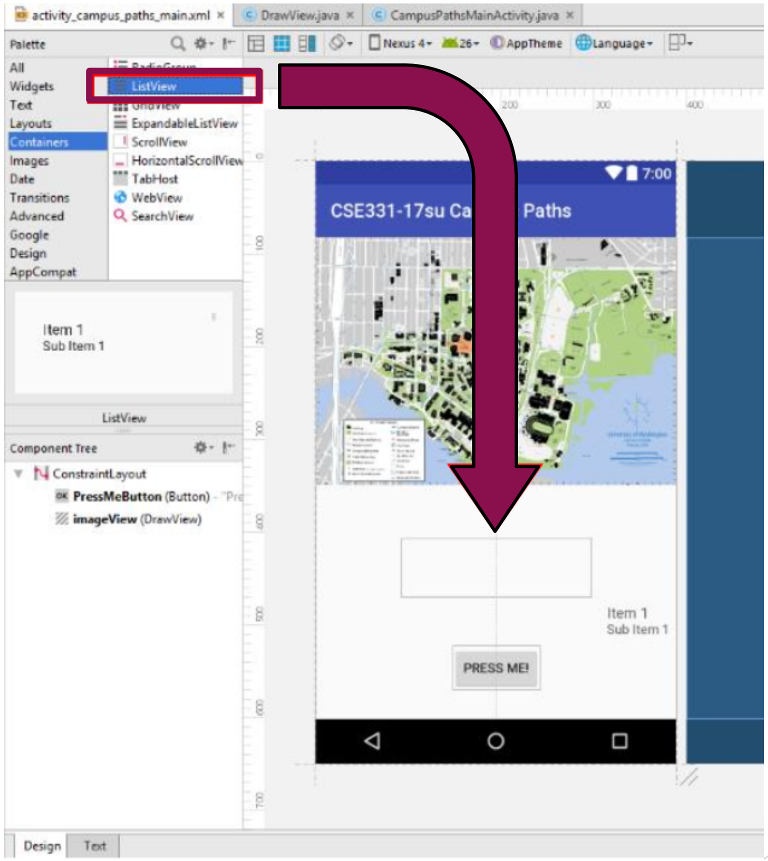
- Extend AppCompatImageView class
- Override "onDraw" method to customize what is displayed on the image
  - Don't forget to call "super.onDraw" first!
- This defines the logic of what will be displayed on your ImageView
- Explicitly request to update the ImageView by calling "invalidate()" on an instance of the widget


activity_campus_paths_main.xml ×
DrawView.java ×
CampusPathsMainActivity.java ×
DrawView
toggleDrawCircle()

```java
package uwid.cse331_17sucampuspaths;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.support.v7.widget.AppCompatImageView;
import android.util.AttributeSet;


public class DrawView extends AppCompatImageView {

    public DrawView(Context context) {
        super(context);
    }

    public DrawView(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public DrawView(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        Paint paint = new Paint();
        paint.setColor(Color.RED);

        canvas.drawCircle(50.f, 50.f, 50.f, paint);
    }
```

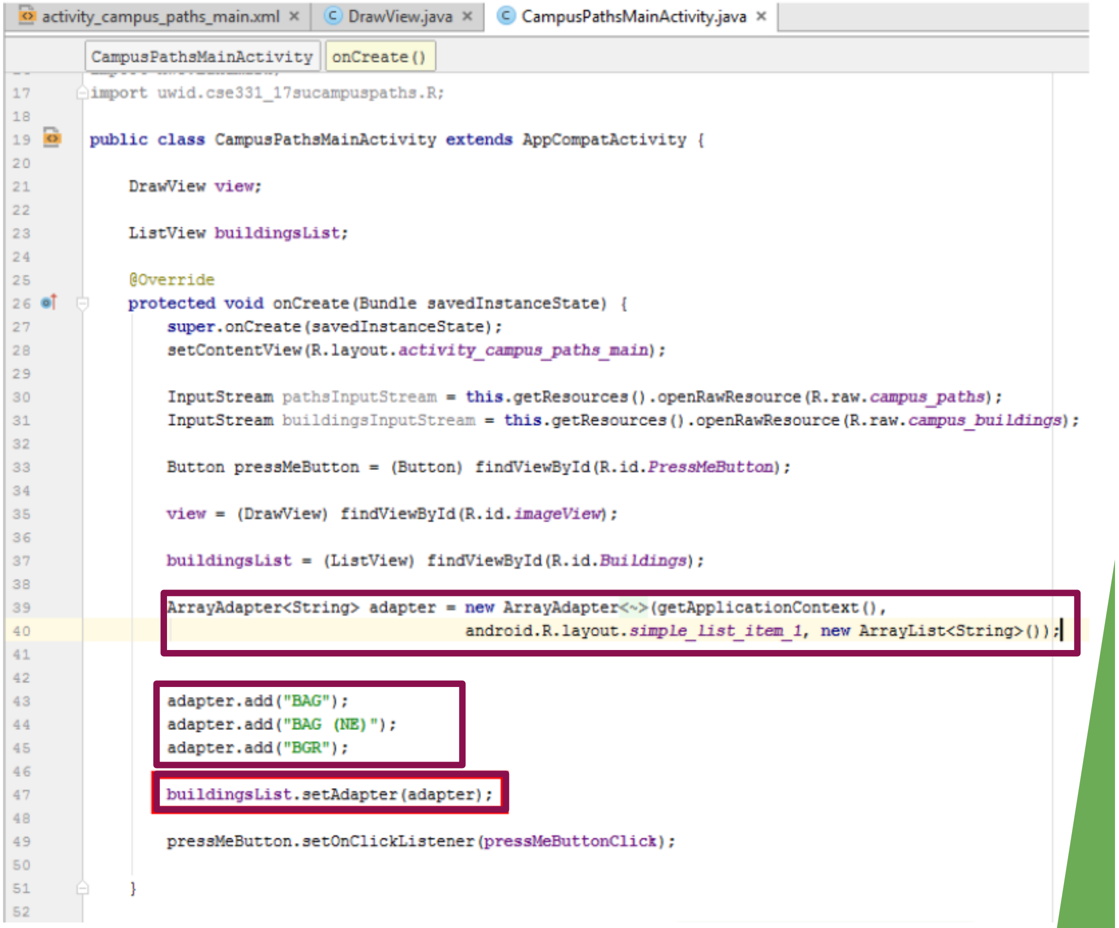
Add Lists to Application

- Navigate to the “Design” tab again
- Drag-and-drop a ListView like the other widgets into the View



Add elements to the ListView

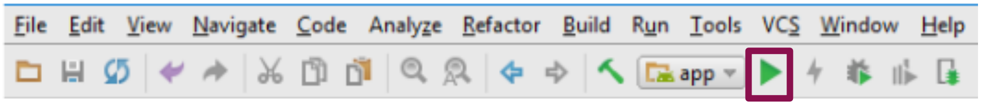
- Requires using an Adapter to populate with elements
    - Functions similar to a Java collection, but designed to function with the ListView
- Load building and path data files from the "raw" resource directory
    - Android requires that these are loaded via an InputStream

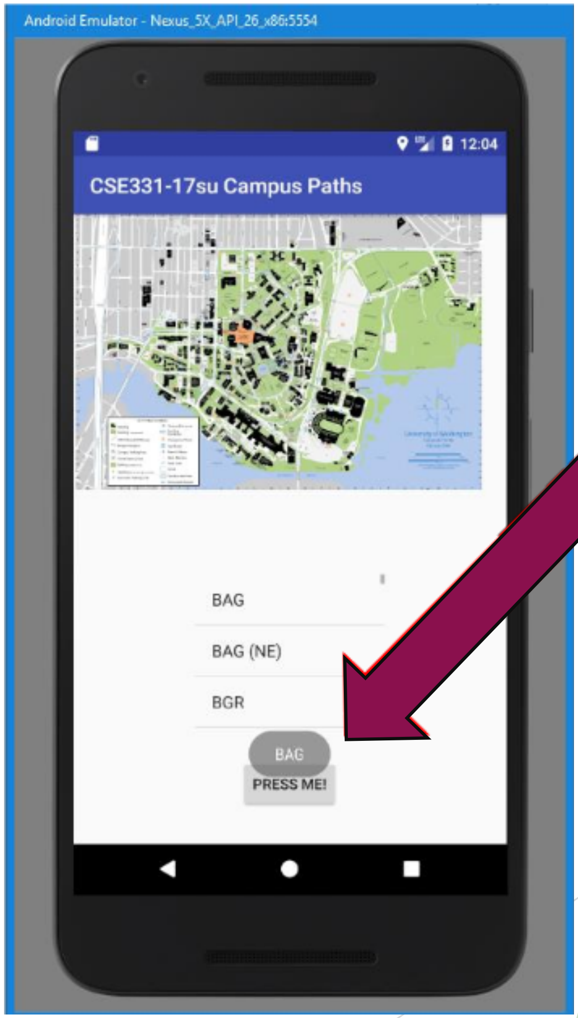

activity_campus_paths_main.xml ×
DrawView.java ×
CampusPathsMainActivity.java ×
CampusPathsMainActivity
onCreate()

```java
import uwid.cse331_17sucampuspaths.R;

public class CampusPathsMainActivity extends AppCompatActivity {

    DrawView view;

    ListView buildingsList;

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

        InputStream pathsInputStream = this.getResources().openRawResource(R.raw.campus_paths);
        InputStream buildingsInputStream = this.getResources().openRawResource(R.raw.campus_buildings);

        Button pressMeButton = (Button) findViewById(R.id.PressMeButton);

        view = (DrawView) findViewById(R.id.imageView);

        buildingsList = (ListView) findViewById(R.id.Buildings);

        ArrayAdapter<String> adapter = new ArrayAdapter<~>(getApplicationContext(),
                                                android.R.layout.simple_list_item_1, new ArrayList<String>());


        adapter.add("BAG");
        adapter.add("BAG (NE)");
        adapter.add("BGR");

        buildingsList.setAdapter(adapter);

        pressMeButton.setOnClickListener(pressMeButtonClick);

    }

```

Run your Application

- Press the “Play” button in Android Studio to run and launch your application!





Android Application Development Guide

- For more step-by-step instructions on how to get started, refer to the official Android Application Development Guide for Homework 9: https://courses.cs.washington.edu/courses/cse331/19wi/docs/tools/AndroidDevelopmentGuide.pdf
- Good luck on the homework!