SSE3052: Embedded Systems Practice

Jinkyu jeong
jinkyu@skku.edu
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu

Android Application

- Application Components
- Manifest
- Resources
- Views and Layout Manager

Application Components

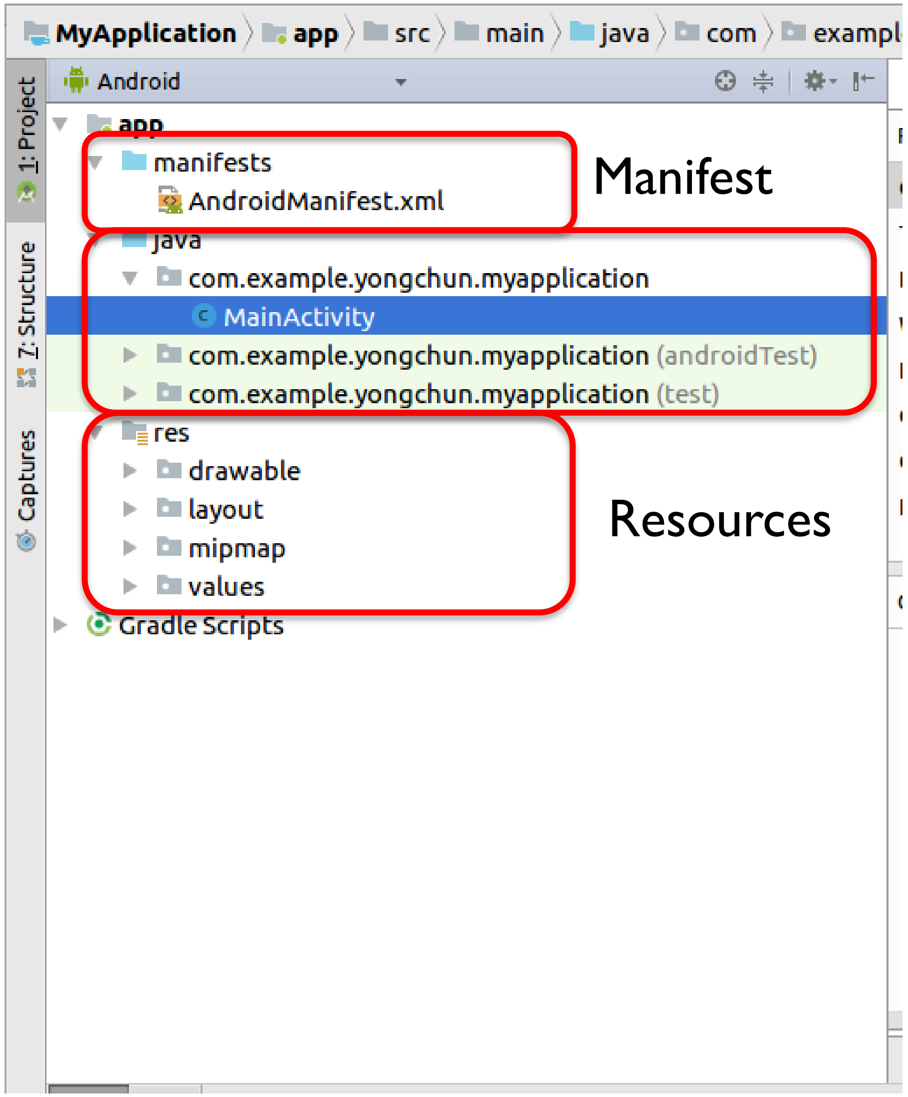
Essential building blocks of an Android app

- Activities
  - Represents a single screen with a user interface
- Services
  - Runs in background to perform long-running operations
- Broadcast Receivers
  - Responds to system-wide broadcast announcements
- Content Providers
  - Manages a shred set of app data
  - Other apps can query the data if content provider allows it

Activating Components

- `Intent` – asynchronous message to request an a ction from other components
- For activities and services, defines the action to pe rform
- For broadcast receivers, defines the announcemen t being broadcast

Project Structure



MyApplication 〉 app 〉 src 〉 main 〉 java 〉 com 〉 exampl

Android

- app
  - manifests
    - AndroidManifest.xml
  - java
    - com.example.yongchun.myapplication
      - MainActivity
    - com.example.yongchun.myapplication (androidTest)
    - com.example.yongchun.myapplication (test)
  - res
    - drawable
    - layout
    - mipmap
    - values
- Gradle Scripts

Manifest

Java source files

Resources

Manifest

- Configuration file, `AndroidManifest.xml`
  - Declares app's components
  - Identifies user permissions the app requires
  - Declares minimum API Level
  - Declares hardware and software features used

Manifest Example

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

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

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

</manifest>
```

- `<application>` defines metadata for application
  - Container for declaring app components
  - `android:icon` attribute points to resources for an icon
  - `@string/app_name` value refers to resource files which contains the actual value of the application name

(cont.)

- `<activity>` defines activity
  - `name` attribute points to class

- `<activity>` elements for activities
- `<service>` elements for services
- `<receiver>` elements for broadcast receivers
- `<provider>` elements for content providers

Resources

- "Resources" that are separate from the source co de
- Example
  - Images (saved in res/drawable/)
  - Audio files
  - Visual presentation of app
    - Animations, menus, styles, colors, and layout of activity

Example: `values/strings.xml`

```xml
<resources>
    <string name="app_name">Hello World</string>
</resources>
```

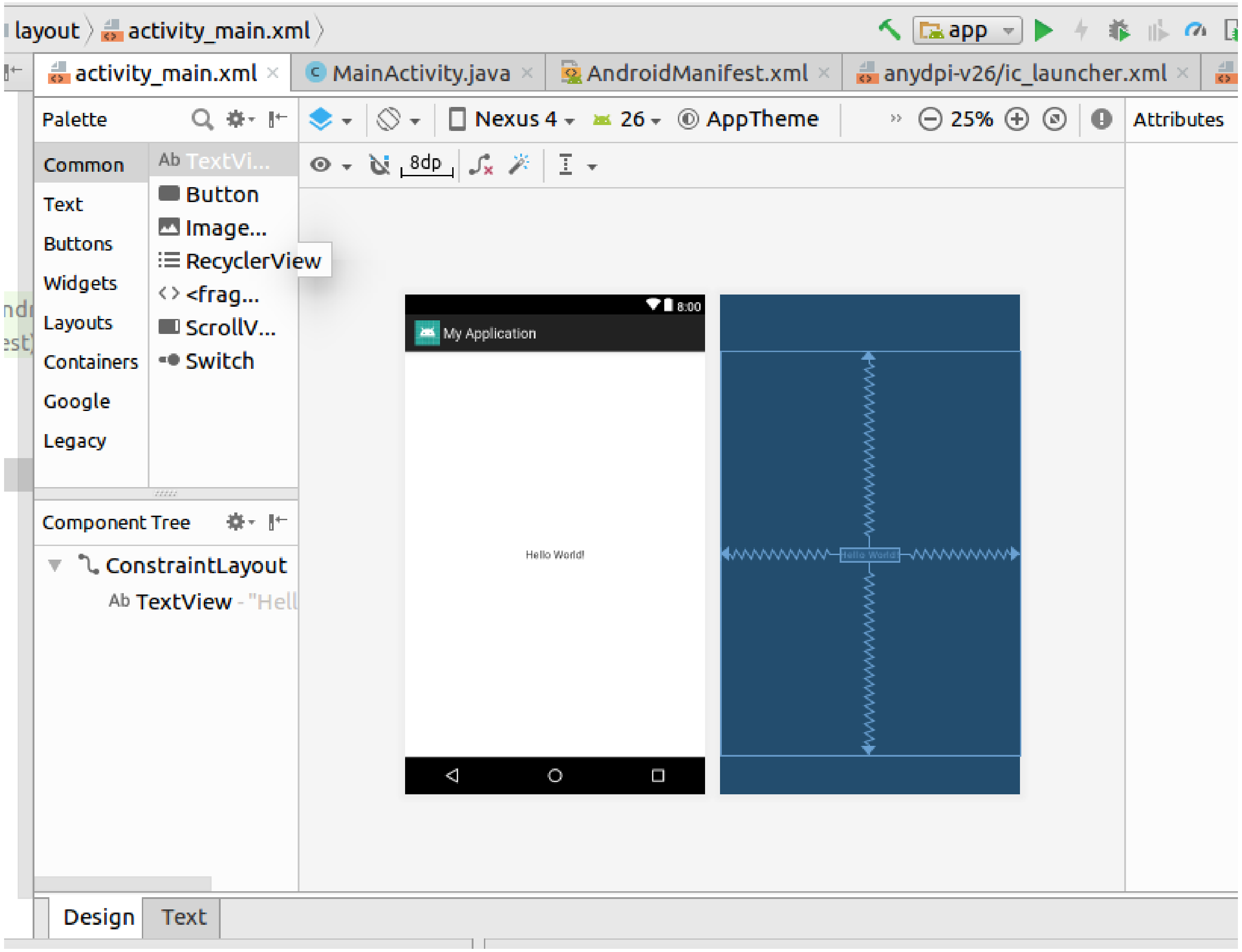
Ex: `layout/activity_main.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="edu.skku.csl.helloworld.MainActivity">


    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />


</RelativeLayout>
```

Ex: `layout/activity_main.xml`



MainActivity.java

```
package edu.skku.csl.helloworld;

import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity {

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

Views and Layout Manager

- View represents a widget
  - Ex) `Button, TextView, EditText` classes
- Layout manager is responsible for layouting itself a nd its child views
  - `LinearLayout`
  - `FrameLayout`
  - `RelativeLayout`
  - `GridLayout`

Accessing Views from Activity

- To access views to access and modify their proper ties
- Use `findViewById(id)` method call

Example

```
<TextView
    android:text="TextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:id="@+id/textView" />
```

Example

```
package edu.skku.csl.helloworld;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView textView = (TextView) findViewById(R.id.textView);
        textView.setText("Bye World!");
        }
}
```

References

- https://developer.android.com/guide/index.html
- https://developer.android.com/reference/packages.html
- http://www.vogella.com/tutorials/Android/article.html

Exercise

- http://www.vogella.com/tutorials/Android/article.html#tutorialtemperature