

MODULE 2

ACTIVITY

TABLE OF CONTENT

- What is an Activity?
- Stack of Activities
- Activity Lifecycle
- Creating an Activity
- Application Object
- Toast Notification
- Webography

What is an Activity?

- ▶ An application = one or more activities
- ▶ Activity
 - Usually represents a single screen
 - Consists of one or more user interface (GUI) controls
 - E.g, TextView, Button, EditText, ...
 - Enables user interaction with the application

What is an Activity?

- ▶ Separation of the presentation layer (View) from business logic
 - View: Activity XML file
 - Defines the user interface of the application
 - Contains GUI controls
 - Business logic: Activity Java file
 - Contains action code of the GUI controls
 - Event handling
 - Processing of data entered by the user

Stack of Activities

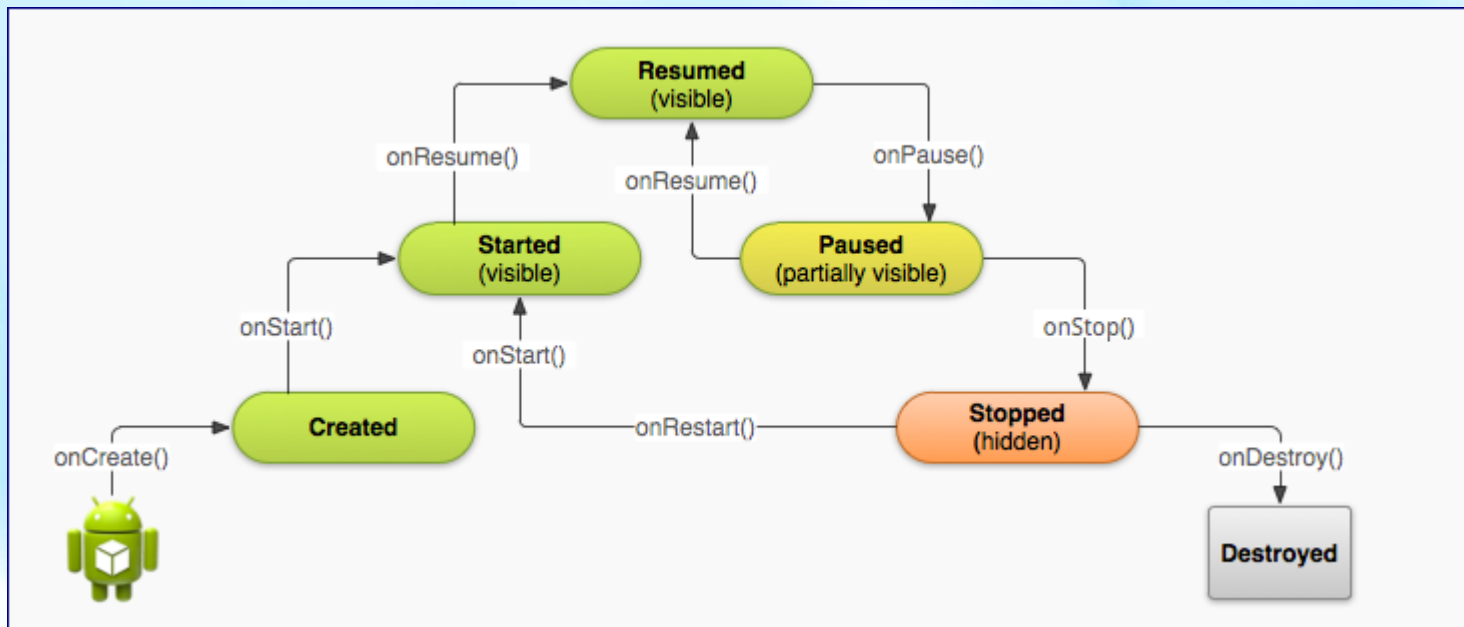
- ▶ Each activity operates independently of the others
- ▶ Stack of activities maintained while running the application
 - The activity on the top is the one currently being displayed
- ▶ When Back button pressed
 - The top activity is popped from the stack
 - ⇒ The previous activity becomes the current activity
 - ⇒ The previous screen is displayed
- ▶ Transition from one activity to another through **intents**
 - Asynchronous message
 - Can be used to pass data from one activity to another one

Activity Lifecycle

▶ Activity States

- Running (visible)
 - Activity is visible and interacts with the user
- Paused
 - Activity is still visible but partially obscured
 - Instance is running but might be killed by the system
- Stopped
 - Activity is not visible
 - Instance is running but might be killed by the system
- Destroyed
 - Activity has been terminated
 - By the system
 - By a call to finish() method

Activity Life Cycle



Source: <http://developer.android.com/training/basics/activity-lifecycle/starting.html>

Creating an Activity

- ▶ Each activity has to be defined in the AndroidManifest.XML

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme"
        android:name="MyApplication"
        >
        <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>

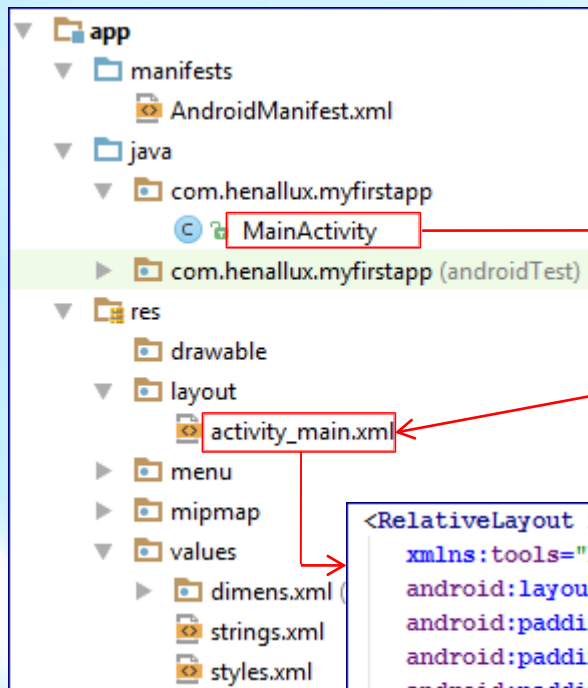
</manifest>
```

→ First launched activity

Creating an Activity

- ▶ Separation of the presentation layer (View) from business logic
- ▶ View: Activity XML file
 - Defines the user interface of the application
 - Contains GUI controls
- ▶ Business logic: Activity Java file
 - Contains action code of the GUI controls
 - Event handling
 - Processing of data entered by the user
 - Subclass of Activity class
 - E.g, AppCompatActivity
 - Base class for activities that use the support library action bar features

Creating an Activity



```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

```
<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>
```

Application Object

- ▶ Create a subclass of the `android.app.Application` class
 - E.g,
 - To store global variables
 - To use preferences
 - To define action to perform when the allocated memory is low
- ▶ Declare this Application subclass in the `AndroidManifest.xml`
- ▶ One object of this class represents the current application
 - It can be retrieved anywhere through `getApplicationContext()` method

Application Object

► E.g,

```
public class MyApplication extends android.app.Application {  
  
    private int generalValue;  
  
    @Override  
    public void onCreate()  
    {    super.onCreate();  
        generalValue = 100;  
    }  
  
    public int getGeneralValue() {  
        return generalValue;  
    }  
}
```

► In AndroidManifest.xml

```
<application  
    android:allowBackup="true"  
    android:icon="@drawable/ic_launcher"  
    android:label="@string/app_name"  
    android:theme="@style/AppTheme"  
    android:name="MyApplication">
```

Application Object

- ▶ Access to the application object

```
MyApplication myApplicationObject = (MyApplication) this.getApplicationContext();  
int dataFromAppObject = myApplicationObject.getGeneralValue();
```

Toast Notification

- ▶ Transient message
 - Automatically disappears after a while without user interaction
- ▶ Used to inform user about non important happenings
 - Not a problem if unnoticed by user
- ▶ Through android.widget.**Toast** class
 - Method : `makeText`
 - Arguments : context (Activity), string to display, duration
 - E.g,

```
Toast.makeText(MainActivity.this, "Welcome!", Toast.LENGTH_SHORT).show();
```

Webography

- ▶ <http://developer.android.com/training/basics/firstapp/index.html>
- ▶ <http://developer.android.com/guide/components/activities.html>