

Global Startup

Meet-up 5: Activity Lifecycle, Intents, and Event Handlers

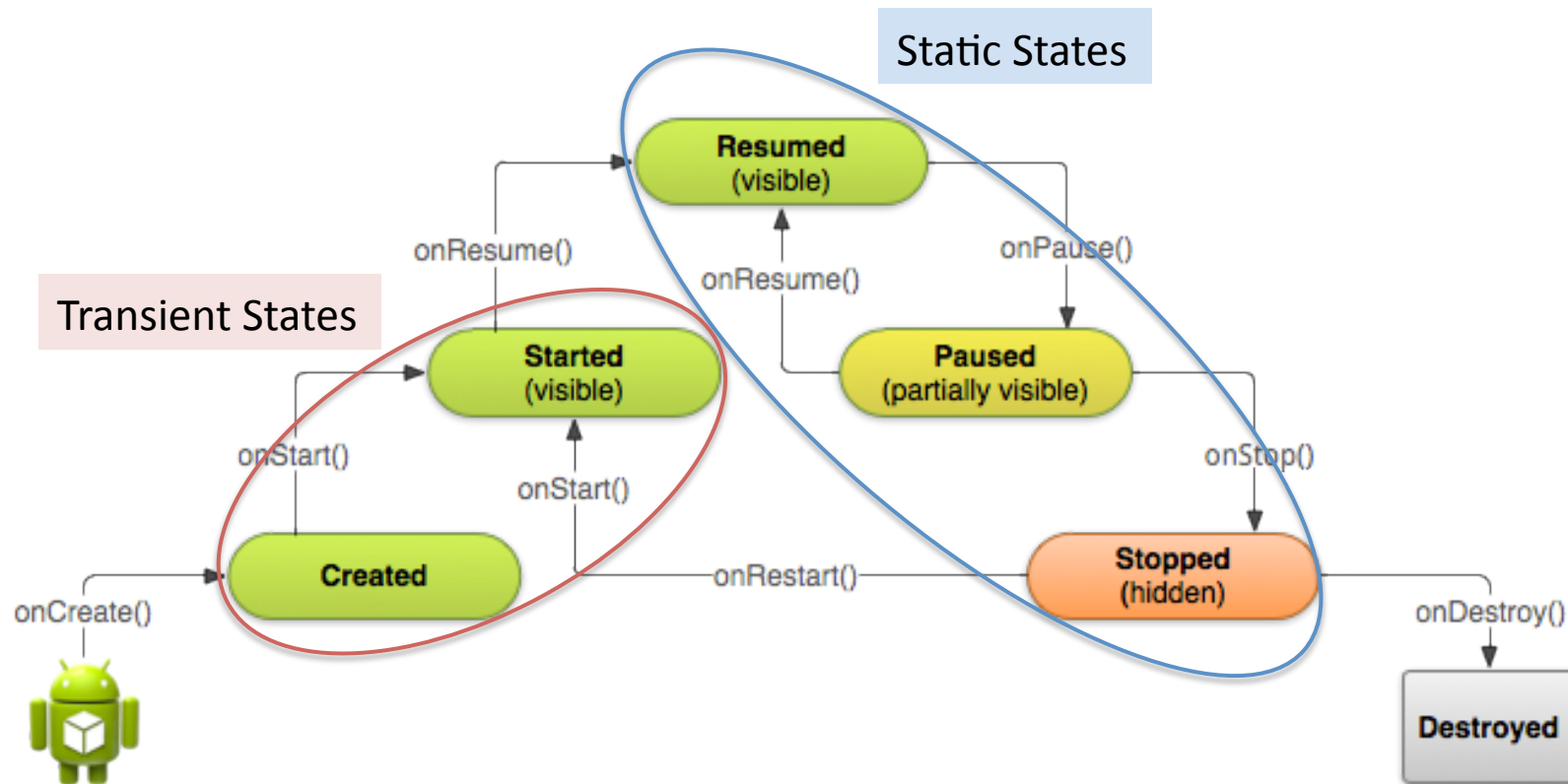


Today's Meet-up

- The Activity Lifecycle
- Intents: Explicit and Implicit
- Event handlers: OnClick, OnLongClick, etc

Activity Lifecycle

Basic Activity Lifecycle



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

onCreate()

Initialize application
(anything that needs to be done only once)

onDestroy()

Clean up anything still outstanding
(usually most cleanup is in onStop)

onStart ()

- Called when your Activity comes back in view
- (Re)start important processes

onStop ()

- Called when switching to new App or Activity
- Save all data, close any connections, free up resources
(Your app might be destroyed!)

onResume ()

- Called when your Activity comes into view
- Do opposite of onPause()
(Create everything)

onPause ()

- Called when your Activity is hidden
- Kill CPU intensive operations, free system resources

Specify State Transition

We have to specify what happens when our activity enters different states

```
Java Code  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
}  
  
public void onPause() {  
    //define what happens if activity enters the pause state  
    //e.g. stop camera  
}  
  
public void onResume() {  
    //define what happens if activity enters the resume state  
    //e.g. initialise camera  
}
```



onCreate(): UI is inflated from XML code



onPause(): Stop actions/ release resources



onResume(): Start actions/ Initialise resources

For most simple activities onStop(), OnRestart(), onStart() don't have to be implemented

Intents

A photograph of a blue tent with yellow trim pitched in a dark forest at night. A small campfire is burning brightly to the right of the tent, casting a warm glow. The background is filled with dark, dense foliage.

**EVER HAVE SEX WHILE
CAMPING?**

IT'S F*KING INTENTS**

Switching between Activities

Step 1: Define all Activities in your App in the AndroidManifest.xml file

```

Main Activity {
  <application android:name = ".MyApplication" android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".OneActivity"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    Second Activity {
      <activity android:name=".AnotherActivity" android:label="picture capture">
      </activity>
    }
  </application>

```

Step 2: Switch from Main Activity to the activity defined in **AnotherActivity.class**, using Intent objects.

```
Intent intent = new Intent(this, AnotherActivity.class);
startActivity(intent);
```

Passing data between Activities

in your current activity, create an intent

```
Intent i = new Intent(getApplicationContext(), ActivityB.class);  
i.putExtra(key, value);  
startActivity(i);
```

then in the other activity, retrieve those values.

```
Bundle extras = getIntent().getExtras();  
if(extras !=null) {  
    String value = extras.getString(key);  
}
```

key: String

value: any primitive (int, String,
double, String[],...)

Explicit vs. Implicit Intent

Explicit Intent

- Specifies the exact recipient activity or application
- Add additional information to intent
- Example: starting another activity

Method called when button is clicked

```
public void new_activity(View v){  
    Intent intent = new Intent(this, CalculatorActivity.class);  
    //Can define information to pass on to new activity  
    startActivity(intent);  
}
```

Java Code

Recipient specified

Implicit Intents

- Recipient not explicitly specified: Android needs to infer from the other parameters where to pass the intent onto

Example: opening a webpage

```
Uri webpage = Uri.parse("http://www.android.com");  
Intent webIntent = new Intent(Intent.ACTION_VIEW, webpage);
```

← Recipient (Browser)
not specified
BUT: property
webIntent defined

Example: starting a phone call

```
Uri number = Uri.parse("tel:5551234");  
Intent callIntent = new Intent(Intent.ACTION_DIAL, number);
```

← Recipient (Phone) not
specified
BUT: property
callIntent defined

Other ways to exchange data between Activities

	Persistent	Non-Persistent
Primitive	SharedPreferences	Intent
Non-Primitive	Files, SQL database	public static fields, Application class

Event Handlers

Basic Event Handlers

We want to define what happens when we perform an action:

- Click on a button (short or long)
- Press a Key on our Android Device
- Touch, gestures
- etc

onClick()

Can be implemented in XML and called in Java

Java Code

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.RelativeLayout;
import android.content.Intent;

public class MainActivity extends Activity {

    RelativeLayout background;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        background= (RelativeLayout) findViewById(R.id.background);
    }

    public void whiteClicked(View v){
        background.setBackgroundResource(R.color.white);
    }
}
```

XML Code

```
<Button
    android:id="@+id/buttonwhite"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="132dp"
    android:text="@string/white"
    android:textColor="@color/white"
    android:onClick="whiteClicked"/>
```



Called when
buttonwhite is
clicked

Implement EventListeners

For other events an `EventListener` has to be implemented:

- `onClick()`
- `onLongClick()`
- `onFocusChange()`
- `onKey()`
- `onTouch()`
- `onCreateContextMenu()`

Example: onLongClick ()

```
public class MainActivity extends Activity implements OnLongClickListener {
```

```
    RelativeLayout background;  
    Button b1;  
    Button b2;
```

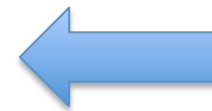


Initiate variables

```
@Override
```

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

```
    b1 = (Button) findViewById(R.id.button1);  
    b2 = (Button) findViewById(R.id.button2);  
    b1.setOnLongClickListener(this);  
    b2.setOnLongClickListener(this);
```



Assign EventListener to
Button widgets

```
    background = (RelativeLayout) findViewById(R.id.background);
```

```
}
```

```
public boolean onLongClick(View v) {
```

```
    switch (v.getId()) {  
        case R.id.button1:  
            background.setBackgroundResource(R.color.White);  
            break;  
        case R.id.button2:  
            background.setBackgroundResource(R.color.Black);  
            break;
```



Define action triggered
by our event

```
}
```

```
return false;
```

```
}
```

Java Code

Today's Assignment

Finish TicTacToe

- Implement the functionality
- Let us know when you're finished – We'll give you an extra challenge

