

# Assignment 02

# COS30017 - Software Development for Mobile Devices

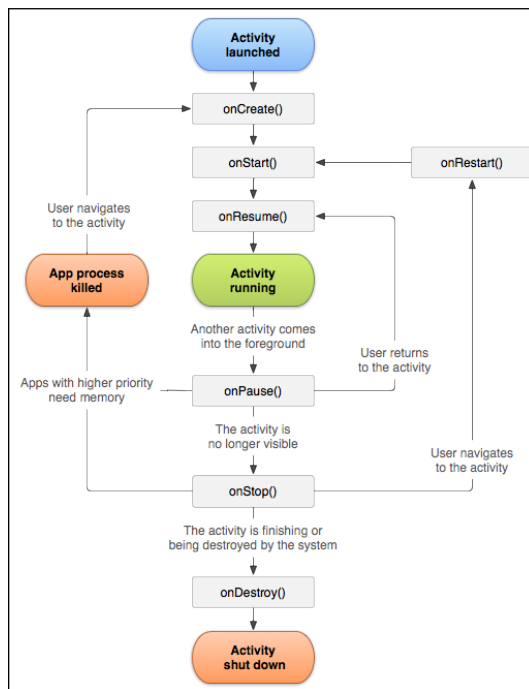
Daniel Parker 971328X

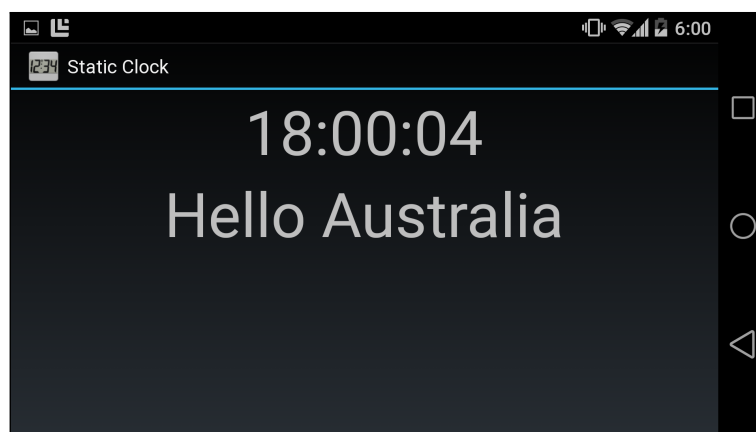
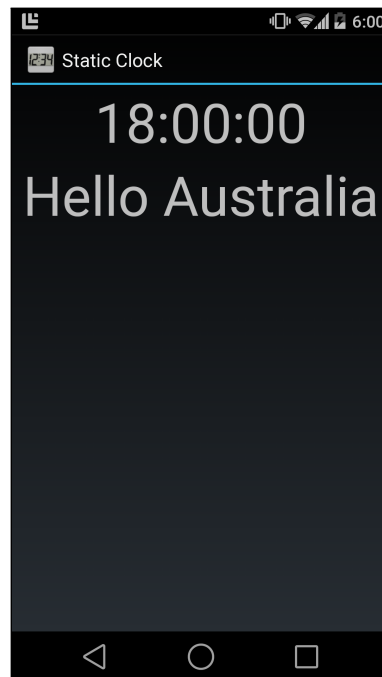
September 1, 2014

## 1. Task 1

### 1.1. Orientation Changes

When the device orientation changes, a certain sequence of steps is being followed by the activity's lifecycle. Before the orientation of the device changes the activity is running, once the phone has been turned to it's side the activity moves to *onPause()* then *onStop()* and then *onDestroy()*. The activity is then created again at *onCreate()* with the new orientation and follows the lifecycle from there onwards as usual.





We observe a time change after the orientation change because when the activity is destroyed and then recreated at *onCreate()* the actual time is updated to the time at which it was recreated.

## 1.2. Resume, Pause and Stop States

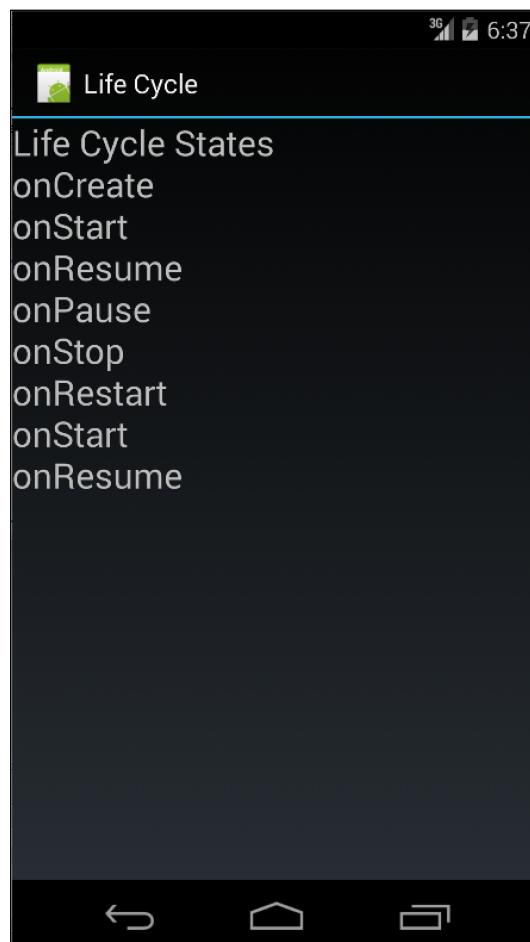
The *Pause* state is entered when the user navigates away from the specific activity by clicking a settings button (thus entering another activity) or otherwise. The activity has not been destroyed it has just moved into the pause state.

When the user navigates back to the activity within the app, the activity will then enter the *Resume* state to prepare for the user to start using it again. This transition happens most commonly when the back button is pressed within an app and the activity comes back into the foreground.

If the user exits the activity either by navigating away from the app or returning to a previous activity then this activity will enter the *Stop* state once it is no longer visible. The activity will pass the *Pause* state on it's way to this state.

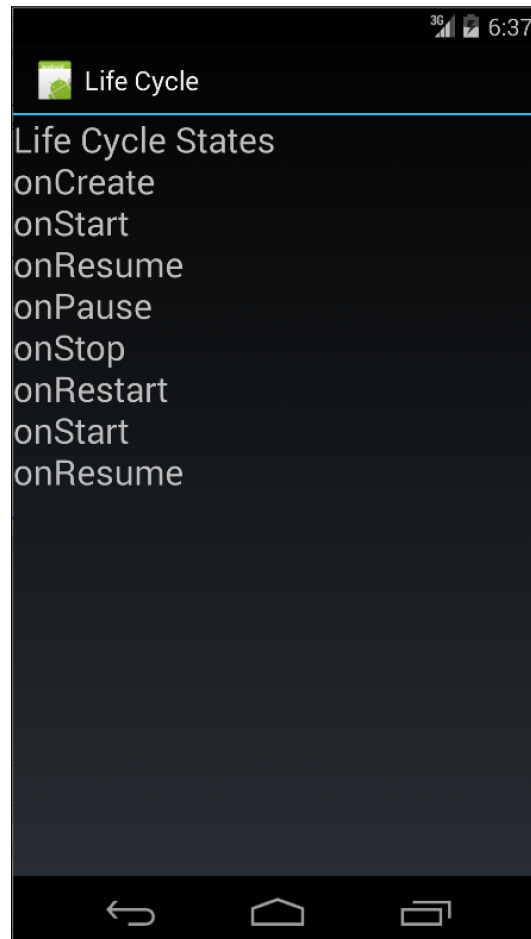
## 2. Task 2

As shown in the screenshot below, the *onRestart()* event can be triggered. This was done by pressing the 'Home' button and then navigating back to the app using the app drawer.



### 3. Task 3

Invoking all the different lifecycle states we can see those that are printable in the activity normally, however the *onDestroy()*— state never gets printed to the activity because the activity is no longer visible on the screen when the state occurs and due to the activity being destroyed the text won't be visible when the activity starts again because the data is lost.



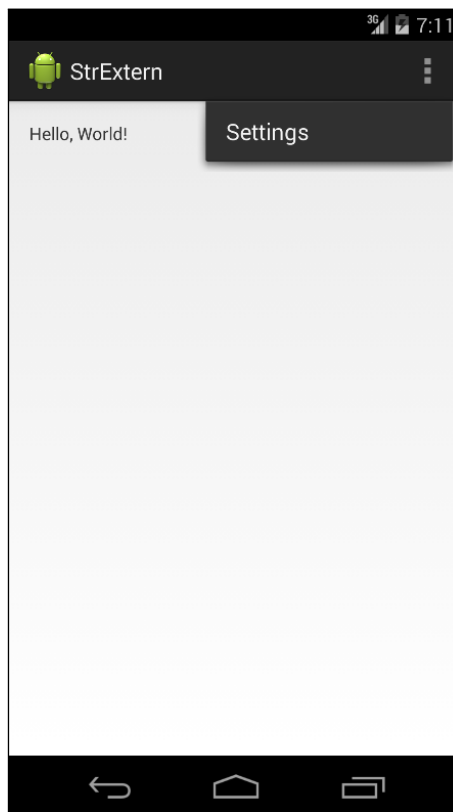
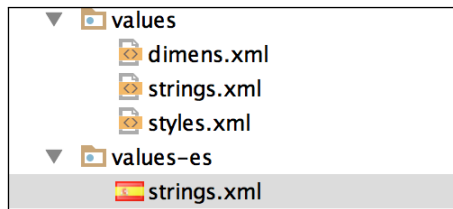
The following screenshots show the logcat messages for when each state is reached.

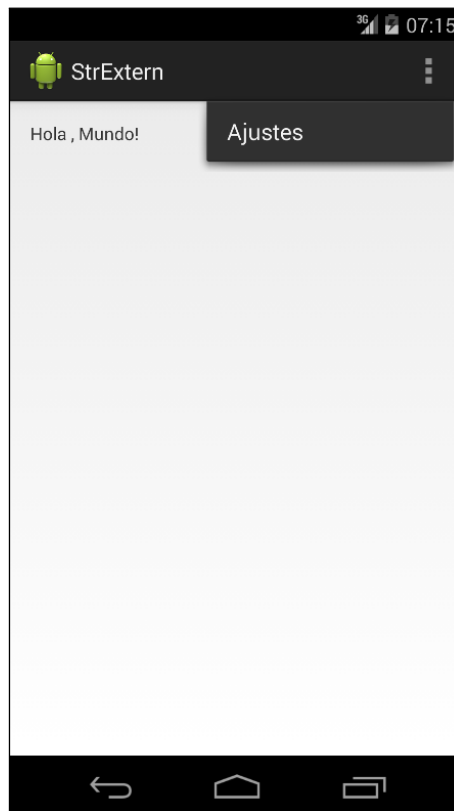
09-01 06:48:11.939	1294-1294/? I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE : onCreate
09-01 06:48:11.939	1294-1294/? I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE : onStart
09-01 06:48:11.949	1294-1294/? I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE : onResume
09-01 06:48:19.849	1294-1294/? I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE : onPause
09-01 06:48:20.959	1294-1294/? I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE : onStop

09-01 06:53:56.939	1390-1390/? I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE : onDestroy
--------------------	---

## 4. Task 4

String externalisation allows us to supply different language strings for the same reference string. For example in this Hello, World! program there is an English strings.xml and a Spanish strings.xml. The Android convention puts them in two different values directories within *res*. The English strings.xml are just in the normal *values* directory, and the Spanish strings.xml are in the *values-es* directory which is short for *Espanol*.





#### 4.1. Source

```
<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>
```

## **5. Task 5**