

Lab Session 4 – Android activity life cycle

.Objectives

The objectives for this week are:

- Understand the process of the Android activity lifecycle.
- Progress the Arkanoid game to include additional activities including the outline of a splash screen, a settings screen and a main game view.
- Begin to implement an OO approach to the construction of the Arkanoid game.

.Tasks

Task 1

- It is important to understand the process that is used in the Activity lifecycle within Android. In order to do this, we will add in Toast messages to the activity that has been created to view the process that it goes through.
- Open the Arkanoid game that you started in the lab from week 3.
- You should have a basic project that contains a **MainActivity** in the Java code folder in the package that is defined.
- Add in a Toast message in the onCreate method. The basic form of a Toast message is:

```
Toast.makeText( this , "OnPause" , Toast.LENGTH_LONG ).show( ) ;
```

- This is the context taken from the activity, the string is the message displayed, the last variable is the length of time that the message is displayed.
- Review the below to get more information about Toast messages:
 - <http://developer.android.com/guide/topics/ui/notifiers/toasts.html>
 - <http://developer.android.com/reference/android/widget/Toast.html>
- Add in also an onPause() and onResume method to your activity. Add Toast messages to these so that they report when they are entered.

- If you need details of these, reference week 3's lecture.

Task 2

- Additional activities can be used for adding elements to a game structure. Key elements of a game are the need to add in both a splash screen and a menu screen.
- To add in an extra activity, we need to create one. Android Studio has an in built function to assist us to do this.
- Right click on your package and go to new -> Activity -> Blank Activity
- Create a new activity called SplashActivity
- We want it to have a draft splash screen view as we will be discussing the use of layouts to enhance the aesthetics in later weeks.
- Add to the **drawable** folder the image. This can be dragged and dropped in or copy and pasted to the folder.
- Using this, we can add an image to the splash screen. We also need to add a button to the screen so we can navigate from here to the next screen. The form should be:
- ```
<ImageView
 android:id="@+id/arkanoid_image"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_alignParentTop="true"
 android:src="@drawable/arkanoid"
 android:layout_centerVertical="true"
 android:layout_centerHorizontal="true" />

<Button
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="@string/start_button"
 android:id="@+id/button"
 android:layout_centerVertical="true"
 android:layout_centerHorizontal="true" />
```
- This should be added to the Activity\_Splash.xml
- The highlighted section makes reference to the image in the drawable folder. **Note that the file type is not needed.** Android can use jpeg, png and gif files.
- We will look further at layouts and adapting the style of the appearance of the game in later weeks.

## Mobile Games

- Making reference to Week 3's lecture, add an onClick to the button and a suitable method in the activity so that it will move to the MainActivity.
- The button references a *start\_button string*. This must also be added to the appropriate place.
- To be able to run the SplashActivity correctly, you will need to adapt the AndroidManifest.xml
- This file controls all of the parameters for the game structure including the activities that are included and the order that they are run.
- To launch from the SplashActivity, you will need to alter the AndroidManifest so that it resembles:
- ```
<activity
    android:name=".SplashActivity"
    android:label="@string/title_activity_splash" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN"
    />

        <category
    android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```
- Each additional activity that is created must be added to the manifest in order to run. Android Studio manages this process for you. We must, however, indicate which activity is run first. In the above xml, we are highlighting through the intent-filter that SplashActivity is the main activity and it will be the launcher.
- For more information about Intents, see <http://developer.android.com/reference/android/content/Intent.html>
- **If you are unsure of this process, ask your lab tutor.**

Task 3

- In your MainActivity, below the text that has been formed, create a menu button system.
- The buttons should be:

Mobile Games

- **Begin Game**
- **Settings**
- **Exit**
- Create a new **GameActivity**. This will be where the “Begin Game” button directs to.
- As with the previous section, add a suitable onClick method to the Begin Game.
- You now should have a game that flows as:



Additional Tasks

- Create a **SettingsActivity** that is accessed by the Settings button from the MainActivity.
- Using both the wysiwyg and the XML for the Activity_Main.xml, adjust the parameters of the buttons so that they are suitably aligned down the middle of the screen and below the text.
- Make the XML as efficient as possible. For reference, look at the attached for assistance: <http://developer.android.com/guide/topics/ui/layout/relative.html>