**MIND STORM SOFTWARE PVT LTD**

**Hands-On Exercise – ex03.doc**

**Objective**Multiple Activities in an Android Application

**Assumptions**

* Development Environment for Android (Java SDK, Eclipse, Android SDK) has been setup successfully.
* You are familiar with using Eclipse.
* Android SDK 4.x is available and Android Virtual Devices are already created.
* Start the Android Virtual Device to save time.
  + Click on **Window 🡪 Android SDK and AVD Manager**.
  + Select an **Android 4.x** compatible AVD and click on **Start**
  + Select **Scale display to real size** and provide a **Screen Size (in)** as **5 inches** or any other appropriate size for your development machine.

**Step by Step Instructions**

**Step 1 – Create the Android Project**

1. Create a new project. Click on **File🡪New 🡪 Android Application Project**
2. Enter **Project Name or Application Name** as **ex03.** Click on **Next.**
3. Enter **Package Name** as **com.mindstorm.ex03**
4. Leave the Build SDK at the latest version i.e. 4.x.
5. Deselect (Uncheck) the **Create custom launcher icon.** Click on **Next.**
6. In **Create Activity**, go with the default options i.e. go with Create Activity and Blank Activity as selected. Click on **Next.**
7. On the **New Blank Activity,** set the Activity Name to **Activity1** and change the Layout Name to **main**
8. Click on **Finish**
9. (Optional): Verify that the Project runs in your Emulator by **Right-click** the **Project** and **Run As 🡪 Android Application**

**Step 2 – Create a Second Activity**

1. Go to **Package Explorer** in Eclipse for the **ex03** project.
2. Select the package **com.mindstorm.ex03** from within the **src** directory.
3. **Right Click** and select **New 🡪 Class**
4. Enter the name as **Activity2**
5. Enter the **Superclass** as **android.app.Activity**
6. Click on **Finish.**

**Step 3 – Code the Main Activity**

1. First we will add some strings to the resources which will be references in the activities. Go to **res/values** folder and modify **strings.xml** to have the following additional string elements.

<string name=*"activity1"*>I am Activity 1</string>

<string name=*"activity2"*>I am Activity 2</string>

<string name=*"launchActivity2"*>Launch Activity 2</string>

1. Define the Layout for **Activity1**. This will consist of a button which will launch the second activity i.e. **Activity2**

**First, let us delete the main.xml file since we do not need it.**

Right-click on **res/layout** , select **New 🡪 File** and provide the name as **activity1.xml**. Enter the following content in the **activity1.xml** file (You can simply copy this):

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:orientation=*"vertical"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

>

<TextView android:text=*"@string/activity1"* android:gravity=*"center\_horizontal"* android:id=*"@+id/textView1"* android:textAppearance=*"?android:attr/textAppearanceLarge"* android:layout\_width=*"fill\_parent"* android:layout\_height=*"wrap\_content"*></TextView>

<TextView android:text=*"Please click button below to sign in first"* android:id=*"@+id/textView2"* android:layout\_width=*"wrap\_content"* android:layout\_height=*"wrap\_content"*></TextView>

<Button android:layout\_height=*"wrap\_content"* android:id=*"@+id/btnLaunchActivity2"* android:text=*"@string/launchActivity2"* android:layout\_width=*"match\_parent"*></Button>

</LinearLayout>

1. Go to **src** 🡪 **com.mindstorm.ex03** and open **Activity1.java** file
2. Modify the **onCreate()** method as show below:

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity1*);

Button btnLaunchActivity2 = (Button) findViewById(R.id.*btnLaunchActivity2*);

btnLaunchActivity2.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View v) {

Intent i = **new** Intent(getBaseContext(),Activity2.**class**);

startActivity(i);

}

});

}

**Step 4 – Code the Activity2**

1. Define the Layout for the **Activity2**. Go to Right-click on **res/layout** , select **New 🡪 File** and provide the name as **activity2.xml**. Enter the following content in the **activity2.xml** file:

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:orientation=*"vertical"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

>

<TextView android:text=*"@string/activity2"* android:gravity=*"center\_horizontal"* android:id=*"@+id/textView1"* android:textAppearance=*"?android:attr/textAppearanceLarge"* android:layout\_width=*"fill\_parent"* android:layout\_height=*"wrap\_content"*></TextView>

<TextView android:text=*"UserName"* android:id=*"@+id/textView2"* android:layout\_width=*"wrap\_content"* android:layout\_height=*"wrap\_content"*></TextView>

<EditText android:layout\_height=*"wrap\_content"* android:id=*"@+id/editText1"* android:layout\_width=*"match\_parent"*>

</EditText>

<TextView android:text=*"Password"* android:id=*"@+id/textView3"* android:layout\_width=*"wrap\_content"* android:layout\_height=*"wrap\_content"*></TextView>

<EditText android:layout\_height=*"wrap\_content"* android:id=*"@+id/editText2"* android:layout\_width=*"match\_parent"*></EditText>

<Button android:text=*"Login"* android:id=*"@+id/button1"* android:layout\_width=*"fill\_parent"* android:layout\_height=*"wrap\_content"*></Button>

</LinearLayout>

1. Go to **src** 🡪 **com.mindstorm.ex03** and open **Activity2.java** file
2. Modify the source code as shown below for **Activity2.java**:

**package** com.mindstorm.ex03;

**import** android.app.Activity;

**import** android.os.Bundle;

**public** **class** Activity2 **extends** Activity {

/\*\* Called when the activity is first created. \*/

@Override

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity2*);

}

}

1. **Step 5 – Update the Manifest**

We need to ensure that both Activity1 and Activity2 are defined in the AndroidManifest.xml file. Go to AndroidManifest.xml and add the following the following <activity> elements inside of the <application> tag as shown below.

<activity android:name=*".Activity1"*

android:label=*"@string/app\_name"*>

<intent-filter>

<action android:name=*"android.intent.action.MAIN"* />

<category android:name=*"android.intent.category.LAUNCHER"* />

</intent-filter>

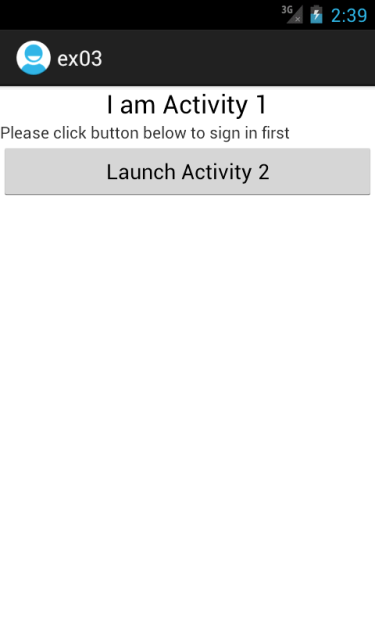
</activity>

<activity android:name=*".Activity2"* android:label=*"@string/app\_name"*></activity>

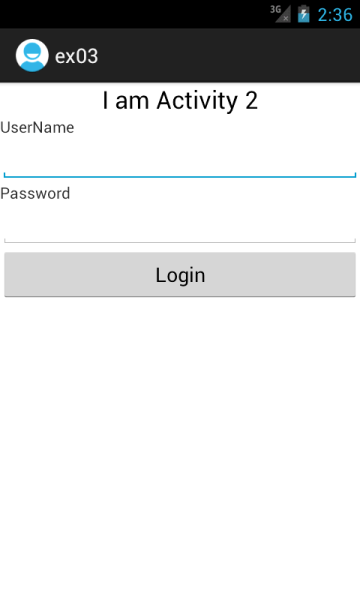
**Step 6 – Run the Example**

1. **Right Click** the **Project** in Eclipse.
2. Select **Run As 🡪 Android** Application
3. (Optional): If you have multiple compatible AVDs running, select the correct AVD. In our case it is the 4.x AVD.

You should see the Activity1 screen come up as shown below:



On clicking the Launch Activity 2 button, you should see the second activity screen come up as shown below:



**Summary**

This hands-on exercise demonstrated how you can have multiple activities (screens) in your Android application. The pattern is simple. Define the Activity class and its layout. Ensure that the Activities are defined in the **AndroidManifest.xml** file. And launch the activity by explicitly providing the Activity class in the Intent and starting it with **startActivity** method.