Copyright:

Author: Daniel DANG

School of Computing (SoC), Eastern Institute of Technology (EIT)

Email: daniel.dang.nz@gmail.com

Lab 6: Make ACPM App Fancy/Professional by Using Coverflow & Pop-up Menu

Objectives

Students have developed a "typical" and "standard" Animation College Programme Management (**ACPM**) App in previous tutorial. In this practical lab, students learn to make ACPM app more fancy and professional by **using CoverFlow library**.

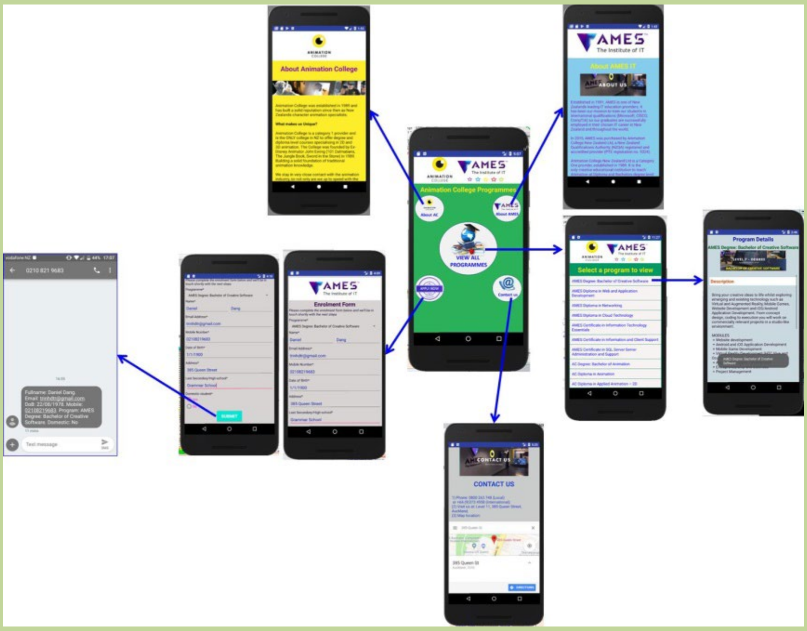
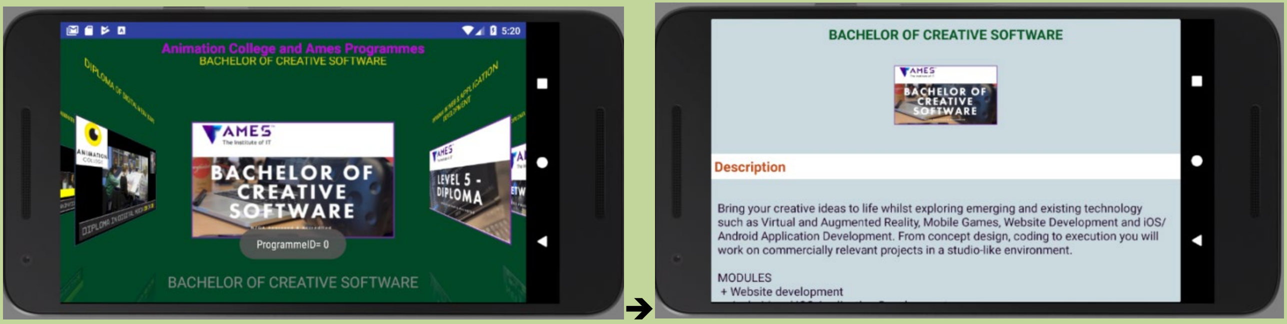


Figure 1: Typical and standard ACPM App

Cover-flow Layout is an exciting topic in interface designing. It can act like a **ViewPager** with swiping to change content view, but also, the center item (selected one) is larger than another. This view style gives a 3D effect, easy to manipulate with the elements.


5:20
Animation College and Ames Programmes
BACHELOR OF CREATIVE SOFTWARE
AMES
BACHELOR OF CREATIVE SOFTWARE
LEVEL 5 - DIPLOMA
ProgrammeID= 0
BACHELOR OF CREATIVE SOFTWARE
➔
BACHELOR OF CREATIVE SOFTWARE
Description

Bring your creative ideas to life whilst exploring emerging and existing technology such as Virtual and Augmented Reality, Mobile Games, Website Development and iOS/ Android Application Development. From concept design, coding to execution you will work on commercially relevant projects in a studio-like environment.

MODULES
+ Website development

In this tutorial, students implement a powerful library to m**ake a flow (carousel) layout for "View all Programmes" by using third-party library FeatureCoverFlow**.

Part 1: Analyze and design "carousel layout"
Add dependencies to Gradle

Step 1: Create a new Android Project

- Application name: **ACPM_Coverflow_[yourname]**
- Company Domain: **ac.ames.project.[yourname]**
- Target Android devices: **Phone and Tablet**;
  - Minimum SDK: **API 21: Android 5.0**
- Add an Activity to Mobile: **Empty Activity**;
  - Activity name: **MainActivity**
  - Layout name: **activity_main**

+ Target devices: run on 71.3% devices
_Smartphone & Tablet
_Android KitKat (5.0) & API21 (min)

Step 2: Collect and prepare all resources (images & texts) for this feature:

+ Collect titles of 12 programmes from websites:

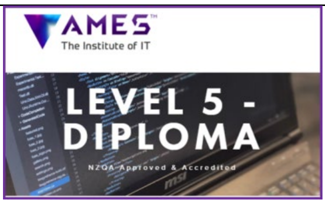
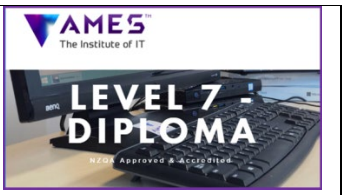
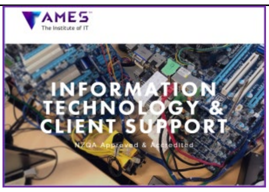
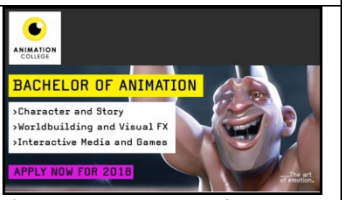
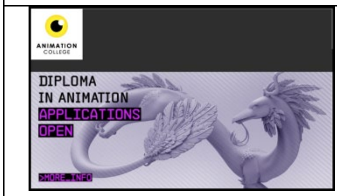
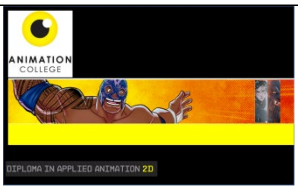
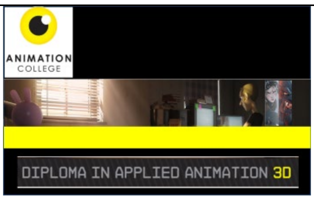
All 12 programmes titles

(1) AMES Degree: Bachelor of Creative Software
(2) AMES Diploma in Web and Application Development
(3) AMES Diploma in Networking
(4) AMES Diploma in Cloud Technology
(5) AMES Certificate in Information Technology Essentials
(6) AMES Certificate in Information and Client Support
(7) AMES Certificate in SQL Server Administration and Support
(8) AC Degree: Bachelor of Animation
(9) AC Diploma in Animation
(10) AC Diploma in Applied Animation – 2D
(11) AC Diploma in Applied Animation – 3D
(12) AC Diploma in Digital Media

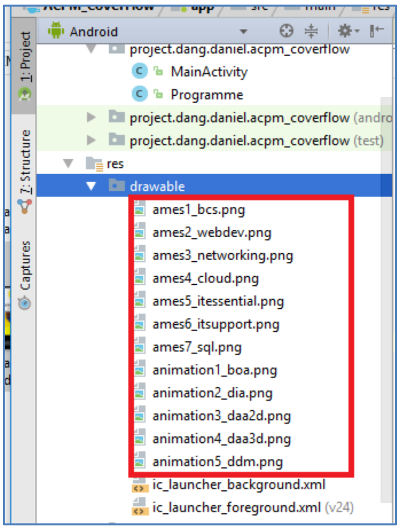
+ Then we declare 1 String variable ("all_programmes_array") in strings.xml file:
Open strings.xml file in "values" folder and add 1 variables "all_programmes_array":

```
<!--Add "all_programmes_array" array variable that contains 12 programmes of AC and AMES-->
<string-array name="all_programmes_array">
    <item>AMES Degree: Bachelor of Creative Software</item>
    <item>AMES Diploma in Web and Application Development</item>
    <item>AMES Diploma in Networking</item>
    <item>AMES Diploma in Cloud Technology</item>
    <item>AMES Certificate in Information Technology Essentials</item>
    <item>AMES Certificate in Information and Client Support</item>
    <item>AMES Certificate in SQL Server Administration and Support</item>
    <item>AC Degree:  Bachelor of Animation</item>
    <item>AC Diploma in Animation</item>
    <item>AC Diploma in Applied Animation - 2D</item>
    <item>AC Diploma in Applied Animation - 3D</item>
    <item>AC Diploma in Digital Media</item>
</string-array>
```

+ Design 12 illustration images for 12 propgrammes:

AMES The Institute of IT BACHELOR OF CREATIVE SOFTWARE NZQA Approved & Accredited (1) AMES Degree: Bachelor of Creative Software Ames1_ bcs.png	 AMES The Institute of IT LEVEL 5 - DIPLOMA NZQA Approved & Accredited (2) AMES Diploma in Web and Application Development: Ames2_webdev.png	AMES The Institute of IT NETWORKING NZQA Approved & Accredited (3) AMES Diploma in Networking Ames3_networking.png	 AMES The Institute of IT LEVEL 7 - DIPLOMA NZQA Approved & Accredited (4) AMES Diploma in Cloud Technology Ames4_cloud.png
AMES The Institute of IT INFORMATION TECHNOLOGY ESSENTIALS NZQA Approved & Accredited (5) AMES Certificate in Information Technology Essentials Ames5_itessential.png	 AMES The Institute of IT INFORMATION TECHNOLOGY & CLIENT SUPPORT NZQA Approved & Accredited (6) AMES Certificate in Information and Client Support Ames6_itsupport.png	AMES The Institute of IT LEVEL 5 - CERTIFICATE (7) AMES Certificate in SQL Server Server Administration and Support Ames7_sql.png	 ANIMATION COLLEGE BACHELOR OF ANIMATION ›Character and Story ›Worldbuilding and Visual FX ›Interactive Media and Games APPLY NOW FOR 2018 8) AC Degree: Bachelor of Animation Animation1_boa.png
 ANIMATION COLLEGE DIPLOMA IN ANIMATION APPLICATIONS OPEN MORE INFO (9) AC Diploma in Animation Animation2_dia.png	 (10) AC Diploma in Applied Animation – 2D Animation3_daa2d.png	 (11) AC Diploma in Applied Animation – 3D Animation4_daa3d.png	ANIMATION COLLEGE DIPLOMA IN DIGITAL MEDIA 2D OR 3D (12) AC Diploma in Digital Media Animation5_ddm.png

+ Copy 12 above illustration images for each programme into "drawable" folder:



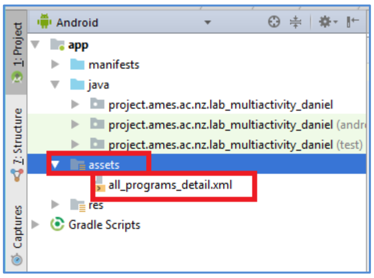
```
project.dang.daniel.acpm_coverflow
    MainActivity
    Programme
project.dang.daniel.acpm_coverflow (andro
project.dang.daniel.acpm_coverflow (test)
res
  drawable
    ames1_bcs.png
    ames2_webdev.png
    ames3_networking.png
    ames4_cloud.png
    ames5_itessential.png
    ames6_itsupport.png
    ames7_sql.png
    animation1_boa.png
    animation2_dia.png
    animation3_daa2d.png
    animation4_daa3d.png
    animation5_ddm.png
    ic_launcher_background.xml
    ic_launcher_foreground.xml (v24)
```

+ Add "Assets" folder to the project:

Within the **"Android" tab** (see the drop-down in the topleft of my image):

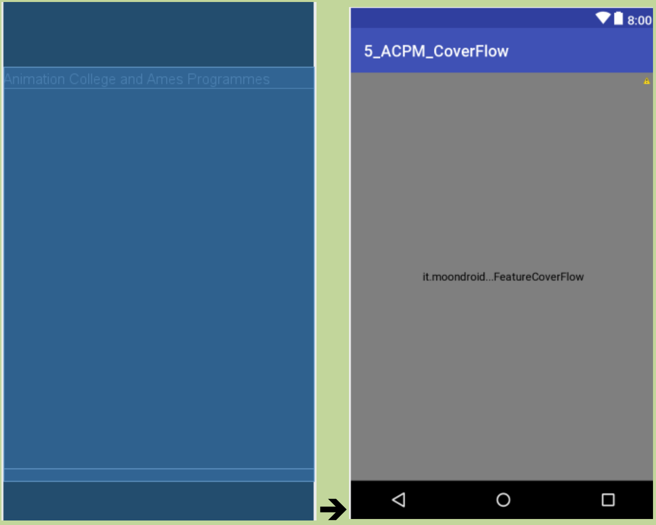
- Right-click on the **"app" folder;**
- **Click: "New" → "Folder" → "Assets Folder"**;
- **A new window appears**: Click **"Finish"**;

+ Copy the all_programs_detail.xml file into the "Assets" folder:


1: Project
2: Structure
Captures
Android
app
manifests
java
project.ames.ac.nz.lab_multiactivity_daniel
project.ames.ac.nz.lab_multiactivity_daniel (andr
project.ames.ac.nz.lab_multiactivity_daniel (test)
assets
all_programs_detail.xml
res
Gradle Scripts

Step 3: Analyze the “main layout”: activity_main.xml

The **main layout** display **all degree, diplomas and certificates** offered by both Animation College and AMES IT (12 in totals: 7 AMES programmes & 5 Animation programmes) in the **form of carousel**.


Animation College and Ames Programmes
8:00
5_ACPM_CoverFlow
it.moondroid...FeatureCoverFlow

So the main layout is very simple, it contains 3 elements:

- A **TextView**: display “**Animation College and Ames Programmes**”;
- 1 **FeatureCoverFlow**: display 12 programmes in form of carousel;
- 1 **TextSwitch**: display the “**programme title**”;

When users selet one programme for viewing more detail, the app will display a “program_detail_layout” showing information: *(1) programme description & modules, (2) NZQF level, (3) duration, (4) start dates, (5) tuition fee, and (6) career opportunities*;

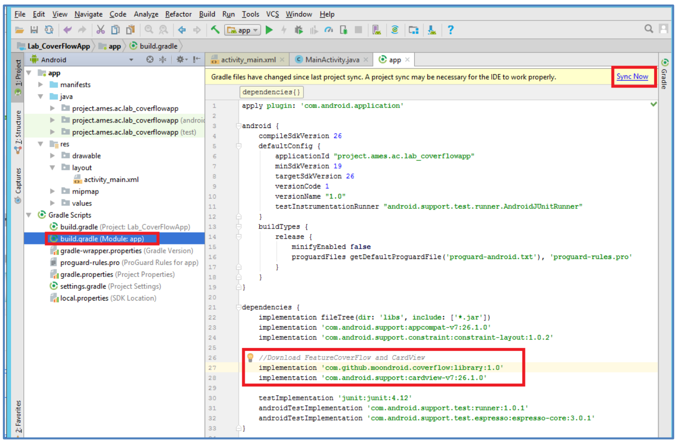
+ Add dependencies to Gradle:

In this tutorial, we will use the third-party element “FeatureCoverFlow” so we need to add dependency to Gradle file. Moreover, we also show a Dialog with a CardView to display each progamme details when it was clicked, we must add dependency of CardView to build.gradle.

Open **build.gradle** and add **FeatureCoverFlow** dependencies and **then “sync” gradle**:

build.gradle:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'

    //Download FeatureCoverFlow and CardView
    implementation 'com.github.moondroid.coverflow:library:1.0'
    implementation 'com.android.support:cardview-v7:26.1.0'

    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
}
```



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
Lab_CoverFlowApp
app
build.gradle
Android
1: Project
7: Structure
Captures
2: Favorites
app
manifests
java
project.ames.ac.lab_coverflowapp
project.ames.ac.lab_coverflowapp (androi
project.ames.ac.lab_coverflowapp (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
build.gradle (Project: Lab_CoverFlowApp)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
MainActivity.java
app
Sync Now
Gradle
dependencies{}

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    defaultConfig {
        applicationId "project.ames.ac.lab_coverflowapp"
        minSdkVersion 19
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'

    //Download FeatureCoverFlow and CardView
    implementation 'com.github.moondroid.coverflow:library:1.0'
    implementation 'com.android.support:cardview-v7:26.1.0'

    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
}
```

+ Open the layout file activity_main.xml and edit it as below:

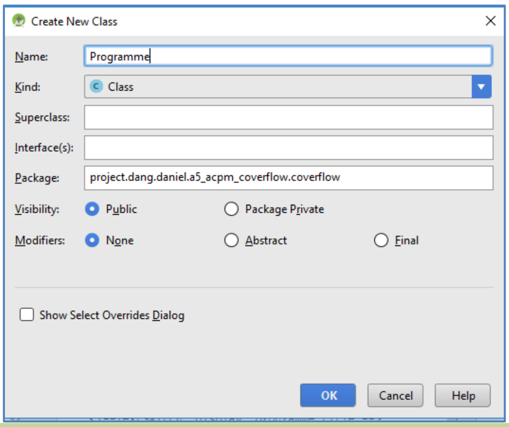
activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#044828"
    tools:context=".MainActivity">

    <!--TextView: display title "Animation College and Ames Programmes"-->
    <TextView
        android:id="@+id/apptitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="Animation College and Ames Programmes"
        android:textColor="#c50fde"
        android:textSize="20sp"
        android:textStyle="bold" />

    <!--FeatureCoverFlow: display programmes in carousel-->
    <it.moondroid.coverflow.components.ui.containers.FeatureCoverFlow
        android:id="@+id/coverFlow"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/apptitle"
        app:coverHeight="250dp"
        app:coverWidth="200dp"
        app:maxScaleFactor="1.5"
        app:reflectionGap="0px"
        app:rotationThreshold="0.5"
        app:scalingThreshold="0.5"
        app:spacing="0.6" />

    <!--TextSwitch: display "programme title"-->
    <TextSwitcher
        android:id="@+id/title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerVertical="true"
        android:paddingBottom="16dp" />
</RelativeLayout>
```

Part 2: Add “Programme” class
Design “programme_view” layout

Now in this part, we add **a “Programme” class** denoting the “**Programme**” properties such as “**programme title**”, “**programme image**”. This “Programme” class represents an “item” in carousel.

Step 1: Add a “Programme” to your project:
Adda new Java class - “Programme” in the same folder containing “MainActivity.java” file:



Create New Class

Name: Programme

Kind: Class

Superclass:

Interface(s):

Package: project.dang.daniel.a5_acpm_coverflow.coverflow

Visibility: ● Public ○ Package Private

Modifiers: ● None ○ Abstract ○ Final

□ Show Select Overrides Dialog

The “Programme” class contains 2 properties:

- Programme title
- Programme image

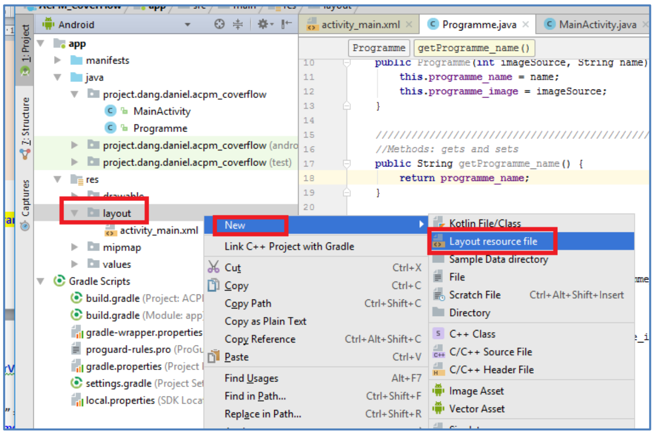
Open Programme.java file as below:

```
public class Programme {
    /////////////////////////////////////////////////////////////////////////
    //Programme properties
    private String programme_title;
    private int programme_image;

    /////////////////////////////////////////////////////////////////////////
    //Constructor
    public Programme(int imageSource, String name) {
        this.programme_title = name;
        this.programme_image = imageSource;
    }

    /////////////////////////////////////////////////////////////////////////
    //Methods: gets and sets
    public String getProgramme_title() {
        return programme_title;
    }

    public int getProgramme_image() {
        return programme_image;
    }

    public void setProgramme_title(String programme_title) {
        this.programme_title = programme_title;
    }

    public void setProgramme_image(int programme_image) {
        this.programme_image = programme_image;
    }
}
```

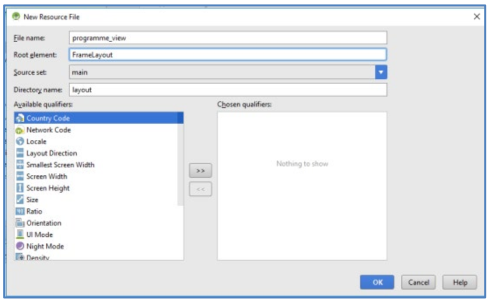
Step 2: Add “layout resource file” (an xml layout) to display “programme”: programme_view.xml

Like **ListView** or **GridView**, **FeatureCoverFlow** is subclass of **AdapterView**, it's also include some **children views**.
We also need an xm layout to display “**programme**” model.

Let’s create an **xml layout** named **programme_view.xml** under “**res**” ⇒ “**layout**” **folder**.
+ First, add a new resource file to “**layout**” folder, called it **programme_view.xml** **(FrameLayout):**


Android
app
manifests
java
project.dang.daniel.acpm_coverflow
MainActivity
Programme
project.dang.daniel.acpm_coverflow (androidTest)
project.dang.daniel.acpm_coverflow (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
build.gradle (Project: ACPM
build.gradle (Module: app
gradle-wrapper.properties
proguard-rules.pro (ProGu
gradle.properties (Project
settings.gradle (Project Set
local.properties (SDK Loca
activity_main.xml
Programme.java
MainActivity.java
Programme
getProgramme_name()
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy as Plain Text
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Kotlin File/Class
Layout resource file
Sample Data directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
1: Project
7: Structure
Captures

```
public Programme(int imageSource, String name)
    this.programme_name = name;
    this.programme_image = imageSource;
}

////////////////////////////////////////////
//Methods: gets and sets
public String getProgramme_name() {
    return programme_name;
}
```

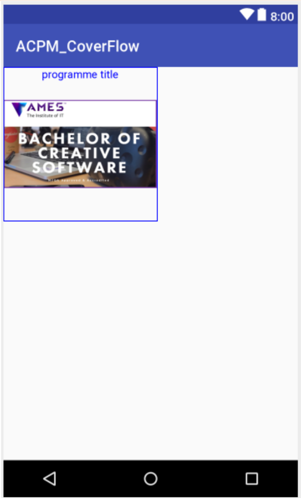


File name: programme_view
Root element: FrameLayout
Source set: main
Directory name: layout

+ Then, open and edit the **programme_view.xml** as below:

programme_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="200dp"
    android:layout_height="200dp">

    <!--ImageView: display programme illustration image-->
    <ImageView
        android:id="@+id/programme_image"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:contentDescription="Item image"
        android:scaleType="fitCenter"
        android:src="@drawable/ames1_bcs" />

    <!--TextView: display programme title-->
    <TextView
        android:id="@+id/programme_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="programme title"
        android:textAppearance="?android:attr/textAppearanceSmallInverse"
        android:textColor="#FF0000FF" />
</FrameLayout>
```

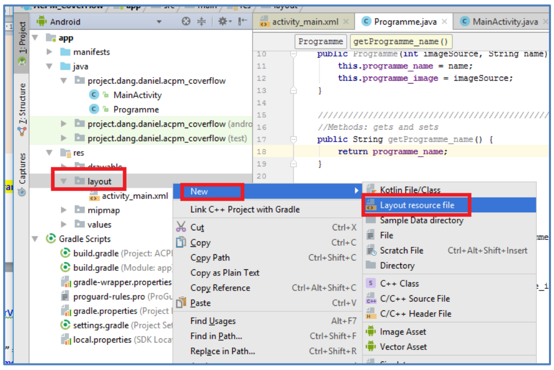


Part 3: Add layout for customizing Dialog layout

Add a layout for customizing Dialog layout: When users click one progrmme, this Dialog view will pop-up and show programme title and its illustration image.

A layout for **customizing Dialog**, only include a **CardView inside**, show when click at each item.

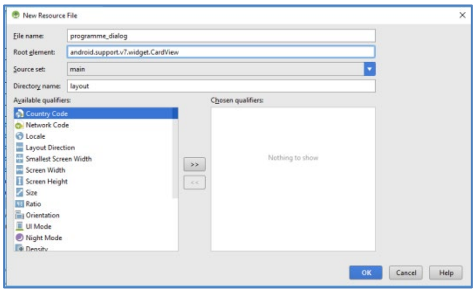
+ First, add a **new resource file** to "**layout**" folder, called it **programme_dialog.xml (type: CardView):**



```
public Programme(int imageSource, String name)
    this.programme_name = name;
    this.programme_image = imageSource;
}

////////////////////////////////////////////
//Methods: gets and sets
public String getProgramme_name() {
    return programme_name;
}
```

Android
activity_main.xml
Programme.java
MainActivity.java
Programme
getProgramme_name()
1: Project
7: Structure
Captures
app
manifests
java
project.dang.daniel.acpm_coverflow
MainActivity
Programme
project.dang.daniel.acpm_coverflow (andro
project.dang.daniel.acpm_coverflow (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
build.gradle (Project: ACPI
build.gradle (Module: app)
gradle-wrapper.properties
proguard-rules.pro (ProGu
gradle.properties (Project
settings.gradle (Project Set
local.properties (SDK Loca
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy as Plain Text
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Kotlin File/Class
Layout resource file
Sample Data directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset

➔



New Resource File

File name: programme_dialog

Root element: android.support.v7.widget.CardView

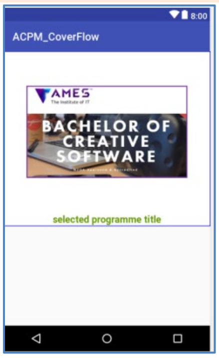
Source set: main

Directory name: layout

+ Then, edit the **programme_dialog.xml** as below:

programme_dialog.xml:

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <!--ImageView: display selected programme illustration image-->
        <ImageView
            android:id="@+id/selectedImage"
            android:layout_width="300dp"
            android:layout_height="300dp"
            android:layout_gravity="center"
            android:contentDescription="Image in Dialog View"
            android:src="@drawable/ames1_bcs" />

        <!--TextView: display selected programme title-->
        <TextView
            android:id="@+id/selectedProgramTitle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="selected programme title"
            android:textColor="@android:color/holo_green_dark"
            android:textSize="18sp"
            android:textStyle="bold" />
    </LinearLayout>
</android.support.v7.widget.CardView>
```



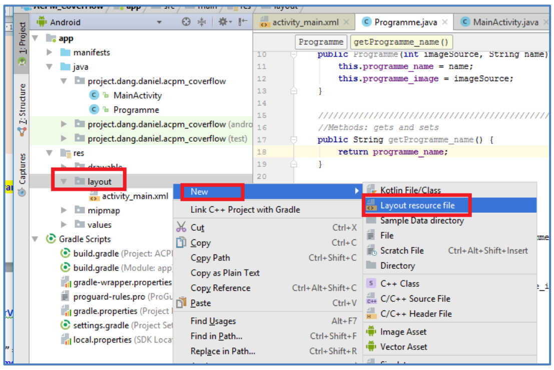
Add a layout for TextSwitcher displaying programme_title to the project

TextSwitcher

In Android, TextSwitcher is a specialized ViewSwitcher that contains only children of type TextView. TextSwitcher is available in Android from version Android 1.6+.

A TextSwitcher is useful to animate a label(i.e. text) on screen. It is an element of transition widget which helps us to add transitions on the labels. Whenever setText(CharSequence) method is called, TextSwitcher simply animates the current text out and new text in.
For Example you need to **cycle through information** in a TextView like Navigating through a list of dates using Left and Right button.

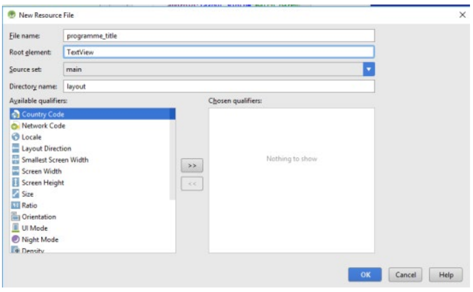
+ First, add a new resource file to "layout" folder, called it programme_title.xml (type: TextView):



```
    public Programme(int imageSource, String name)
        this.programme_name = name;
        this.programme_image = imageSource;
    }

    ///////////////////////////////////////////
    //Methods: gets and sets
    public String getProgramme_name() {
        return programme_name;
    }
```

Android
app
manifests
java
project.dang.daniel.acpm_coverflow
MainActivity
Programme
project.dang.daniel.acpm_coverflow (andro
project.dang.daniel.acpm_coverflow (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
build.gradle (Project: ACP
build.gradle (Module: app
gradle-wrapper.properties
proguard-rules.pro (ProGu
gradle.properties (Project
settings.gradle (Project Set
local.properties (SDK Loca
activity_main.xml
Programme.java
MainActivity.java
Programme
getProgramme_name()
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy as Plain Text
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Kotlin File/Class
Layout resource file
Sample Data directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset

➔



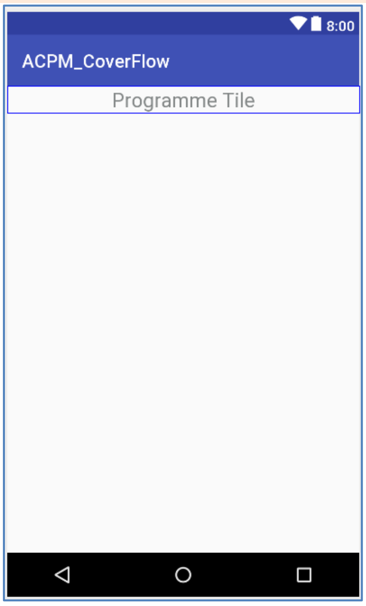
New Resource File

File name: programme_title
Root element: TextView
Source set: main
Directory name: layout

+ Then, edit the pgrogramme_title.xml as below:

programme_title.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textAppearance="?android:textAppearanceLargeInverse"
    android:textColor="#838787"
    tools:text="Programme Title" />
```



Part 4: Add a customized Adapter for FeatureCoverFlow element

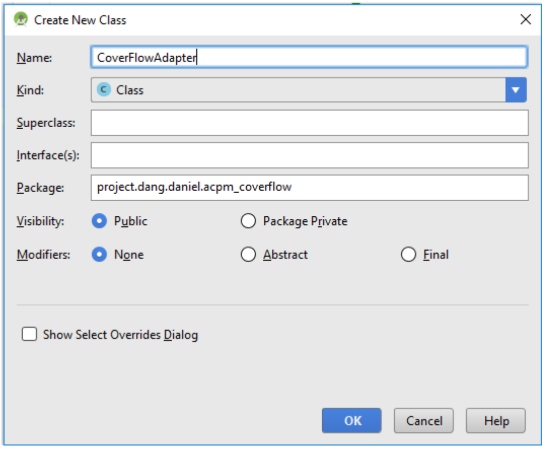
Add a customized Adapter for "FeatureCoverFlow" element

FeatureCoverFlow needs **an adapter** to store and show its children view like ListView.

So, our **customized adapter class** must implement **BaseAdapter**. This customized adapter can be built like a normal **ListView adapter**.

Now let's create a class named **CoverFlowAdapter.java**.

+ First, create a new java class - CoverFlowAdapter:



Create New Class

Name: CoverFlowAdapter

Kind: Class

Superclass:

Interface(s):

Package: project.dang.daniel.acpm_coverflow

Visibility: ● Public ○ Package Private

Modifiers: ● None ○ Abstract ○ Final

☐ Show Select Overrides Dialog

+ Open and edit CoverFlowAdapter.java as below:

CoverFlowAdapter.java:

```
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.ArrayList;

public class CoverFlowAdapter extends BaseAdapter {
    ///////////////////////////////////////////////////////////////////////////////
    //1: Declare variables
    private ArrayList<Programme> programmesList = new ArrayList<>(0);
    private Context mContext;

    ///////////////////////////////////////////////////////////////////////////////
    //2: Constructor
    public CoverFlowAdapter(Context context, ArrayList<Programme> programmesList) {
        this.mContext = context;
        this.programmesList = programmesList;
    }

    ////////////////////////////////////////////////////////////////////////////////
    //3: Add a ViewHolder Class that link to "programme_view.xml" layout
    private static class ViewHolder {
        private TextView programme_title;
        private ImageView programme_image;
    }

    ////////////////////////////////////////////////////////////////////////////////
    //4: Get converted view: Make the link to programme_view.xml layout
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        //
        View rowView = convertView;
        //
        if (rowView == null) {
            //
            LayoutInflater inflater = (LayoutInflater)
mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            rowView = inflater.inflate(R.layout.programme_view, null, false);
            //Assign values to "programme_title" and "programme_image" of viewHolder
            ViewHolder viewHolder = new ViewHolder();
            viewHolder.programme_title = (TextView) rowView.findViewById(R.id.programme_title);
            viewHolder.programme_image = (ImageView) rowView.findViewById(R.id.programme_image);
            //
            rowView.setTag(viewHolder);
        }

        //Assign values to "programme_title" and "programme_image" of holder
        ViewHolder holder = (ViewHolder) rowView.getTag();
        holder.programme_image.setImageResource(programmesList.get(position).getProgramme_image());
        holder.programme_title.setText(programmesList.get(position).getProgramme_title());

        //Return value
        return rowView;
    }

    ///////////////////////////////////////////////////////////////////////////////
    //5: Override getCount() method: return the size of programmesList
    @Override
    public int getCount() {
        return programmesList.size();
    }

    ///////////////////////////////////////////////////////////////////////////////
    //6: Override getItem() method: return which game item has been selected
    @Override
    public Programme getItem(int position) {
        return programmesList.get(position);
    }

    ///////////////////////////////////////////////////////////////////////////////
    //7: Override getItemId() method: return the position of selected item
    @Override
    public long getItemId(int position) {
        return position;
    }
}
```

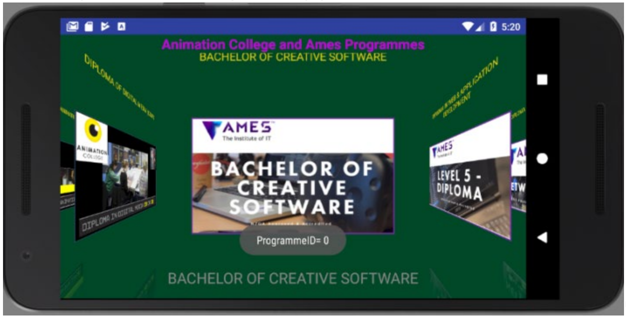
Part 5: Programming MainActivity

+ Open **MainActivity.java** and do the necessary changes:

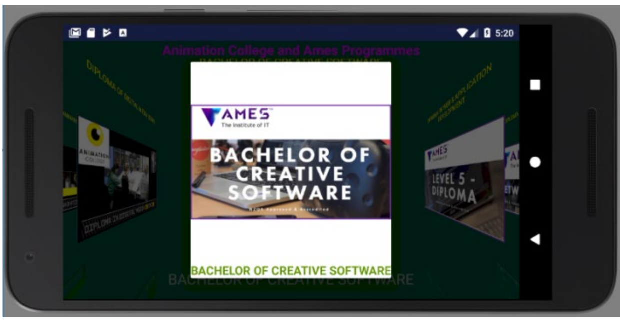
MainActivity.java:

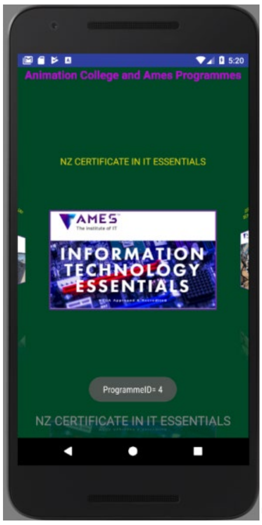
```
import android.app.Dialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ImageView;
import android.widget.TextSwitcher;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ViewSwitcher;

import java.util.ArrayList;

import it.moondroid.coverflow.components.ui.containers.FeatureCoverFlow;

public class MainActivity extends AppCompatActivity {
    ///////////////////////////////////////////////////////////////////////////////////////
    //1: Declare variables
    private FeatureCoverFlow mCoverFlow;
    private CoverFlowAdapter mAdapter;
    private ArrayList<Programme> programmesList;
    private TextSwitcher mTitle;

    ///////////////////////////////////////////////////////////////////////////////////////
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ///////////////////////////////////////////////////////////////////////////////////
        //2: Find reference and do casting
        mCoverFlow = (FeatureCoverFlow) findViewById(R.id.coverFlow);
        //Initialize programmesList
        programmesList = new ArrayList<>();
        //Set Adapter for mCoverFlow
        mAdapter = new CoverFlowAdapter(this, programmesList);
        mCoverFlow.setAdapter(mAdapter);
        //
        mTitle = (TextSwitcher) findViewById(R.id.title);
        mTitle.setFactory(new ViewSwitcher.ViewFactory() {
            @Override
            public View makeView() {
                LayoutInflater inflater = LayoutInflater.from(MainActivity.this);
                TextView textView = (TextView) inflater.inflate(R.layout.programme_title, null);
                return textView;
            }
        });


        //////////////////////////////////////////////////////////////////////////////////
        //3: Initialize few programmesList for testing - will be added below
        prepareProgrammesList();

        //////////////////////////////////////////////////////////////////////////////////
        //4: Setup programmes list
        mAdapter = new CoverFlowAdapter(this, programmesList);
        mCoverFlow.setAdapter(mAdapter);


        //////////////////////////////////////////////////////////////////////////////////
        //5: Set "Item Click" Listener for CoverFlow
        //Set click listener to detect when users click a programme item. When users click a programme item
        //Open up a Dialog View to display "programme details": name & image
        mCoverFlow.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int position, long id) {
                //Pop up the name of selected programme
                Toast.makeText(getApplicationContext(), "Click: " +
programmesList.get(position).getProgramme_title(), Toast.LENGTH_SHORT).show();
                //Pop up a Dialog to display "programme details"
                Dialog dialog = new Dialog(MainActivity.this);
                dialog.setContentView(R.layout.programme_dialog);
                dialog.setCancelable(true); // dismiss when touching outside
                dialog.setTitle("Programme Details");
                //
                TextView text = (TextView) dialog.findViewById(R.id.selectedProgramTitle);
                text.setText(programmesList.get(position).getProgramme_title());
                ImageView image = (ImageView) dialog.findViewById(R.id.selectedImage);
                image.setImageResource(programmesList.get(position).getProgramme_image());
                //
                dialog.show();
            }
        });

        //Set "Scroll" listener for mCoverFlow, will be added below
        //mCoverFlow.setOnScrollPositionListener(onScrollListener());
        mCoverFlow.setOnScrollPositionListener(new FeatureCoverFlow.OnScrollPositionListener() {
            ///
            @Override
            public void onScrolledToPosition(int position) {
                //When the carousel has stopped at a specific position
                mTitle.setText(programmesList.get(position).getProgramme_title());
            }

            ///
            @Override
            public void onScrolling() {
                //When the carousel is scrolling
                mTitle.setText("");
            }
        });
    }

    /////////////////////////////////////////////////////////////////////////////////////
    //6: Add 12 programmes to programmesList
    private void prepareProgrammesList() {
        //Initialize programmesList
        programmesList = new ArrayList<>();

        //Add a programme to the "programmes list"
        Programme programme = new Programme(R.drawable.ames1_bcs, "BACHELOR OF CREATIVE SOFTWARE");
        programmesList.add(programme);

        //Add a programme to the "programmes list"
        programme = new Programme(R.drawable.ames2_webdev, "DIPLOMA IN WEB & APPLICATION DEVELOPMENT");
        programmesList.add(programme);

        //Add item
        programme = new Programme(R.drawable.ames3_networking, "DIPLOMA IN NETWORKING");
        programmesList.add(programme);

        //Add item
        programme = new Programme(R.drawable.ames4_cloud, "DIPLOMA IN CLOUD TECHNOLOGY");
        programmesList.add(programme);

        //Add item
        programme = new Programme(R.drawable.ames5_itessential, "NZ CERTIFICATE IN IT ESSENTIALS");
        programmesList.add(programme);

        //Add item
        programme = new Programme(R.drawable.ames6_itsupport, "CERTIFICATE IN INFORMATION TECHNOLOGY & CLIENT
SUPPORT");
        programmesList.add(programme);

        //Add item
        programme = new Programme(R.drawable.ames7_sql, "CERTIFICATE IN SQL SERVER ADMINISTRATION &
SUPPORT");
        programmesList.add(programme);

        //Add item
        programme = new Programme(R.drawable.animation1_boa, "BACHELOR OF ANIMATION");
        programmesList.add(programme);

        //Add item
        programme = new Programme(R.drawable.animation2_dia, "DIPLOMA IN ANIMATION");
        programmesList.add(programme);
        //Add item
        programme = new Programme(R.drawable.animation3_daa2d, "DIPLOMA IN APPLIED ANIMATION 2D");
        programmesList.add(programme);

        //Add item
        programme = new Programme(R.drawable.animation4_daa3d, "DIPLOMA IN APPLIED ANIMATION 3D");
        programmesList.add(programme);

        //Add item
        programme = new Programme(R.drawable.animation5_ddm, "DIPLOMA OF DIGITAL MEDIA (DDM)");
        programmesList.add(programme);
    }

}
```

+ Compile and run your app on AVD (Nexus 5X):

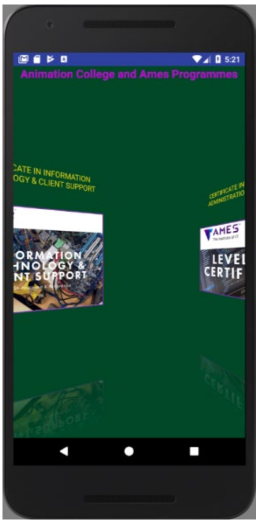

Animation College and Ames Programmes
BACHELOR OF CREATIVE SOFTWARE
ProgrammeID= 0
BACHELOR OF CREATIVE SOFTWARE

→

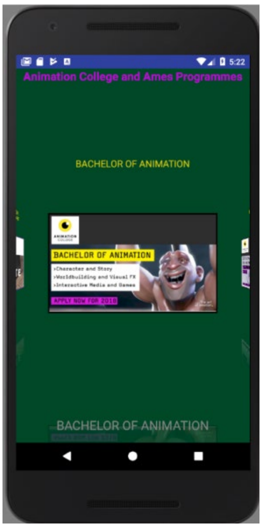



Animation College and Ames Programmes
NZ CERTIFICATE IN IT ESSENTIALS
ProgrammeID= 4
NZ CERTIFICATE IN IT ESSENTIALS

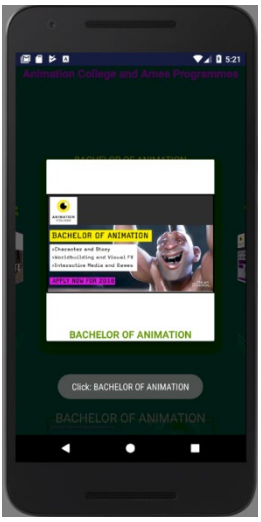
→



→



→



Part 6: Upgrade "Programme Detail Dialog" to show all detailed information

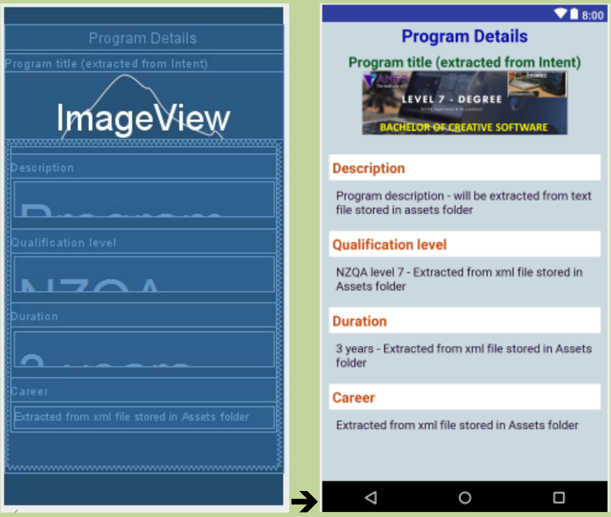
Step 1: Analyze & design the programme_detail_dialog

So far, the program_dialog shows only 2 information: programme title and programme image.
Le't improve this feature: When users click one programme in carousel, the app will open a dialog displaying all detailed information (extracted from all_programs_detail.xml file in "xml" folder) of that programmes.

+ Analyze the programme_detail_dialog.xml layout that contains 10 TextView:

- TextView: display title "Program Details";
- TextView: display title "Program title";
- 1 ImageView : display selected **programme illustration image**;
- Other TextViews: display "program" disruption, qualification level, duration and career opportunities;
- Since the information of each program is quite long so we will use Scrollview and put "Other TextViews" inside;

➔ **The "programme detail dialog" structure look like**:


Program Details
Program title (extracted from Intent)
ImageView
Description
Program
Qualification level
NZQA
Duration
3 years
Career
Extracted from xml file stored in Assets folder
➔
8:00
LEVEL 7 · DEGREE
BACHELOR OF CREATIVE SOFTWARE

Program Details

Program title (extracted from Intent)

Description

Program description - will be extracted from text file stored in assets folder

Qualification level

NZQA level 7 - Extracted from xml file stored in Assets folder

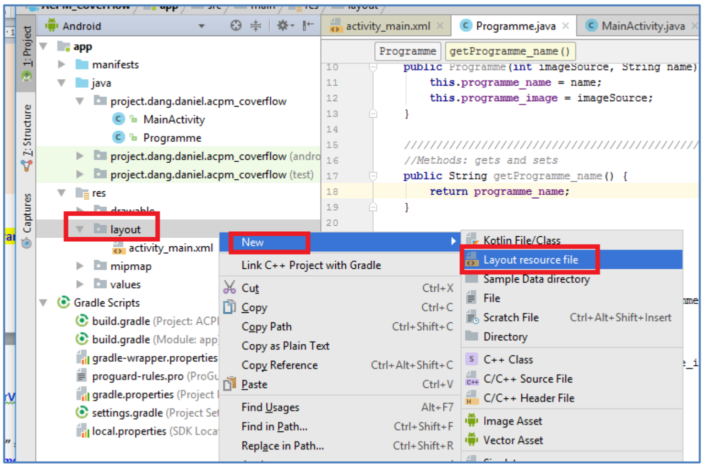
Duration

3 years - Extracted from xml file stored in Assets folder

Career

Extracted from xml file stored in Assets folder

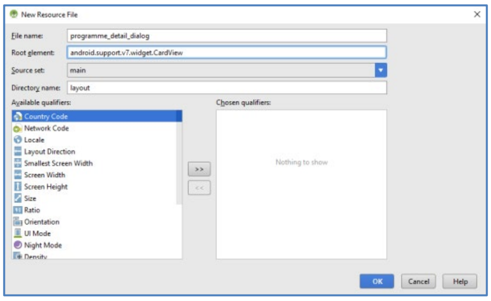
+ First, add a new resource file to “layout” folder, called it programme_detail_dialog.xml (type: CardView):



```
    public Programme(int imageSource, String name)
        this.programme_name = name;
        this.programme_image = imageSource;
    }

    ////////////////////////////////////////////////
    //Methods: gets and sets
    public String getProgramme_name() {
        return programme_name;
    }
```

activity_main.xml
Programme.java
MainActivity.java
Programme
getProgramme_name()
Android
app
manifests
java
project.dang.daniel.acpm_coverflow
MainActivity
Programme
project.dang.daniel.acpm_coverflow (andro
project.dang.daniel.acpm_coverflow (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
build.gradle (Project: ACPI
build.gradle (Module: app
gradle-wrapper.properties
proguard-rules.pro (ProGu
gradle.properties (Project
settings.gradle (Project Se
local.properties (SDK Loca
1: Project
7: Structure
Captures
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy as Plain Text
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Kotlin File/Class
Layout resource file
Sample Data directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset

➔



New Resource File

File name: programme_detail_dialog

Root element: android.support.v7.widget.CardView

Source set: main

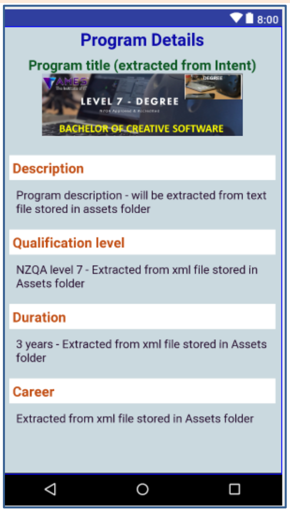
Directory name: layout

+ Then, edit the programme_detail_dialog.xml as below:

programme_detail_dialog.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="#cad9df"
            android:orientation="vertical">

            <!--TextView: display title "Program Details"-->
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_centerHorizontal="true"
                android:gravity="center_vertical|center_horizontal"
                android:padding="2dp"
                android:text="Program Details"
                android:textAlignment="center"
                android:textColor="#0b09a9"
                android:textSize="25sp"
                android:textStyle="bold" />

            <!--TextView: display the name of selected program-->
            <TextView
                android:id="@+id/selected_program"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="5dp"
                android:gravity="center_vertical|center_horizontal"
                android:text="Program title (extracted from Intent)"
                android:textColor="#0c5823"
                android:textSize="20sp"
                android:textStyle="bold" />

            <!--ImageView: display selected programme illustration image-->
            <ImageView
                android:id="@+id/selectedImage"
                android:layout_width="150dp"
                android:layout_height="150dp"
                android:layout_gravity="center"
                android:contentDescription="Image in Dialog View"
                android:src="@drawable/ames1_bcs" />

            <!--TextView: display "Description"-->
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:background="#FFFFFFFF"
                android:padding="5dp"
                android:text="Description"
                android:textColor="#c24d12"
                android:textSize="20sp"
                android:textStyle="bold" />

            <!--TextView: display program description-->
            <TextView
                android:id="@+id/programDescription"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_margin="5dp"
                android:background="@null"
                android:padding="5dp"
                android:text="Program description - will be extracted from text file stored in assets folder"
                android:textColor="#240c2d"
                android:textSize="17sp"
                android:textStyle="normal" />

            <!--TextView: display "Qualification level"-->
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:background="#FFFFFFFF"
                android:padding="5dp"
                android:text="Qualification level"
                android:textColor="#c24d12"
                android:textSize="20sp"
                android:textStyle="bold" />

            <!--TextView: display program qualification level-->
            <TextView
                android:id="@+id/qualification_level"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_margin="5dp"
                android:background="@null"
                android:padding="5dp"
                android:text="NZQA level 7 - Extracted from xml file stored in Assets folder"
                android:textColor="#240c2d"
                android:textSize="17sp"
                android:textStyle="normal" />

            <!--TextView: display "Duration"-->
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:background="#FFFFFFFF"
                android:padding="5dp"
                android:text="Duration"
                android:textColor="#c24d12"
                android:textSize="20sp"
                android:textStyle="bold" />

            <!--TextView: display program duration-->
            <TextView
                android:id="@+id/durationTxt"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_margin="5dp"
                android:background="@null"
                android:padding="5dp"
                android:text="3 years - Extracted from xml file stored in Assets folder"
                android:textColor="#240c2d"
                android:textSize="17sp"
                android:textStyle="normal" />

            <!--TextView: display "Career"-->
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:background="#FFFFFFFF"
                android:padding="5dp"
                android:text="Career"
                android:textColor="#c24d12"
                android:textSize="20sp"
                android:textStyle="bold" />

            <!--TextView: display program career-->
            <TextView
```

```
                android:id="@+id/careerTxt"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_margin="5dp"
                android:background="@null"
                android:padding="5dp"
                android:text="Extracted from xml file stored in Assets folder"
                android:textColor="#240c2d"
                android:textSize="17sp"
                android:textStyle="normal" />
        </LinearLayout>
    </ScrollView>
</android.support.v7.widget.CardView>
```



8:00

Program Details

Program title (extracted from Intent)

Description

Program description - will be extracted from text file stored in assets folder

Qualification level

NZQA level 7 - Extracted from xml file stored in Assets folder

Duration

3 years - Extracted from xml file stored in Assets folder

Career

Extracted from xml file stored in Assets folder

Step 2: Let's program MainActivity so that the app will extract relevant program data stored in all_programs_detail.xml file and display it on programme_detail_dialog

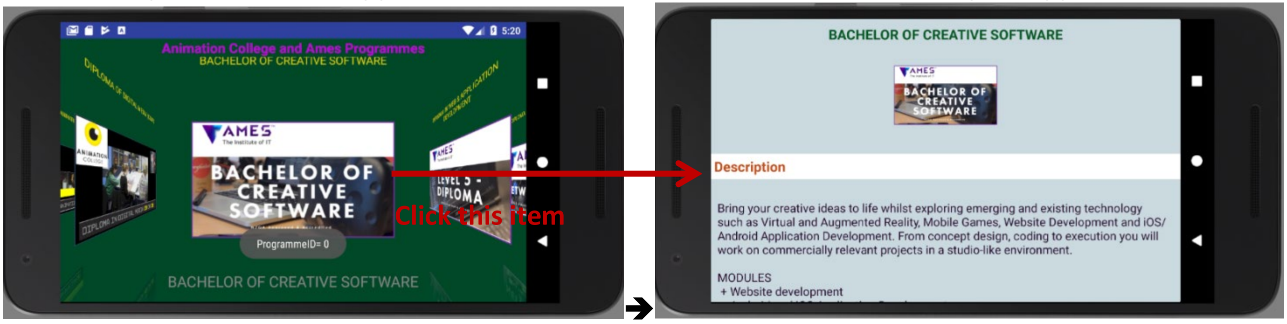
+ Open MainActivity.java, go to setOnClickListener() function for mCoverFlow element and edit it as below:

- Create XMLPullParser object and retrieve the xml file stored in Assets folder to extract data;

```
/////////////////////////////////////////////////////////////////////////////////
//5: Set "Item Click" Listener for CoverFlow
//Set click listener to detect when users click a programme item. When users click a programme item
//Open up a Dialog View to display "programme details": name & image
mCoverFlow.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int position, long id) {
        //Pop up a Dialog to display "programme details"
        //Dialog dialog = new Dialog(MainActivity.this);
        Dialog dialog = new Dialog(MainActivity.this, android.R.style.Theme_Black_NoTitleBar_Fullscreen);
        //dialog.setContentView(R.layout.programme_dialog);
        dialog.setContentView(R.layout.programme_detail_dialog);
        dialog.setCanceledOnTouchOutside(true);
        //------------------------------------------------------------------------
        //Programme title
        TextView selected_program = (TextView) dialog.findViewById(R.id.selected_program);
        selected_program.setText(programmesList.get(position).getProgramme_title());

        //Programme image
        ImageView programmeIllustrationImage = (ImageView) dialog.findViewById(R.id.selectedImage);
        programmeIllustrationImage.setImageResource(programmesList.get(position).getProgramme_image());

        //Other info will be extracted from all_programs_detail.xml
        TextView programDescription = (TextView) dialog.findViewById(R.id.programDescription);
        TextView programQualificationLevel = (TextView) dialog.findViewById(R.id.qualification_level);
        TextView programDuration = (TextView) dialog.findViewById(R.id.durationTxt);
        TextView programCareer = (TextView) dialog.findViewById(R.id.careerTxt);
        //Other "String" variables storing programme details
        String programTitle = "", description = "", qualificationLevel = "", duration = "", career = "";


        String selectedProgrammeTitle =
getResources().getStringArray(R.array.all_programmes_array)[position];
        //4: Extract the relevant information from "all_programs_detail.xml" file by using XML parser
        try {
            //5: Open program_detail.xml file stored in Assets folder
```

```java
            InputStream inputStream = getAssets().open("all_programs_detail.xml");

            //6: Use of XML DOM Parser for extracting data
            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance().newInstance();
            DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
            Document document = documentBuilder.parse(inputStream);
            Element element = document.getDocumentElement();
            element.normalize();

            //7: Read all the nodes containing tag "program"
            NodeList nodeList = document.getElementsByTagName("program");

            //8: Loop through all nodes to find the relevant selected program
            boolean program_found = false;
            for (int i = 0; i < nodeList.getLength(); i++) {
                //9: Get "node" in xml file
                Node node = nodeList.item(i);
                Element sub_Element = (Element) node;
                //9: Get the program "title"
                programTitle = sub_Element.getElementsByTagName("title").item(0)
                        .getChildNodes().item(0).getNodeValue();
                programDescription.setText(programTitle);
                //10: Check if the program title is the selected program. If yes, display its detail
                if (programTitle.contains(selectedProgrammeTitle)) {
                    description = sub_Element.getElementsByTagName("description").item(0)
                            .getChildNodes().item(0).getNodeValue();
                    qualificationLevel = sub_Element.getElementsByTagName("qualification_level").item(0)
                            .getChildNodes().item(0).getNodeValue();
                    duration = sub_Element.getElementsByTagName("duration").item(0)
                            .getChildNodes().item(0).getNodeValue();
                    career = sub_Element.getElementsByTagName("career").item(0)
                            .getChildNodes().item(0).getNodeValue();
                    //11: Change the variable program_found to "true"
                    program_found = true;
                }
            }

            //11: If not found any program in the xml file, assign all variables to "Not found"
            if (!program_found) {
                description = "Not found";
                qualificationLevel = "Not found";
                duration = "Not found";
                career = "Not found";
            }

            //12: Display the extracted program details into the TextView on Layout
            programDescription.setText(description);
            programQualificationLevel.setText(qualificationLevel);
            programDuration.setText(duration);
            programCareer.setText(career);

        } catch (Exception e) {
            e.printStackTrace();
        }

        //-------------------------------------------------------------------------
        dialog.show();
    }
});
```

+ Now, compile and run the app on AVD (Nexus 5X) to observe the behavior of your application:


5:20
Animation College and Ames Programmes
BACHELOR OF CREATIVE SOFTWARE
AMES
BACHELOR OF CREATIVE SOFTWARE
LEVEL 5 - DIPLOMA
ProgrammeID= 0
BACHELOR OF CREATIVE SOFTWARE
Click this item

BACHELOR OF CREATIVE SOFTWARE

Description

Bring your creative ideas to life whilst exploring emerging and existing technology such as Virtual and Augmented Reality, Mobile Games, Website Development and iOS/ Android Application Development. From concept design, coding to execution you will work on commercially relevant projects in a studio-like environment.

MODULES
+ Website development

ProgramDetail_Activity.java

```
public class ProgrammeDetail_Activity extends AppCompatActivity {
    ///////////////////////////////////////////////////////////////////////////////////////////
    //1: Declare variables
    private TextView selected_program;
    private ImageView programmeIllustrationImage;
    private TextView programDescription, programQualificationLevel, programDuration, programCareer;
    //Declare an array of 12 illustration images for 12 programmes
    private int[] illustration_images_array = {R.drawable.illustration_image1,
            R.drawable.illustration_image2, R.drawable.illustration_image3, R.drawable.illustration_image4,
            R.drawable.illustration_image5, R.drawable.illustration_image6, R.drawable.illustration_image7,
            R.drawable.illustration_image8, R.drawable.illustration_image9, R.drawable.illustration_image10,
            R.drawable.illustration_image11, R.drawable.illustration_image12};
    //Other "String" variables storing programme details
    private String programTitle, description, qualificationLevel, duration, career;
    private String program_name;//receive from ListviewProgrammes_Activity
    private int program_id;//receive from ListviewProgrammes_Activity

    ///////////////////////////////////////////////////////////////////////////////////////////
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.programme_detail_layout);

        //2: Find references and do casting for visual elements: make connections between UI and java code
        selected_program = (TextView) findViewById(R.id.selected_program);
        programmeIllustrationImage = (ImageView) findViewById(R.id.programme_illustration_image);
        programDescription = (TextView) findViewById(R.id.programDescription);
        programQualificationLevel = (TextView) findViewById(R.id.qualification_level);
        programDuration = (TextView) findViewById(R.id.durationTxt);
        programCareer = (TextView) findViewById(R.id.careerTxt);

        //3: Extract data stored in "Intent" object sent from ListView_Activity and display it
        //Extract "program_name" attached with "Intent" -> then display it in "selected_program" TextView
        program_name = getIntent().getExtras().getString("program_name");
        selected_program.setText(program_name);
        //Extract "program_id" attached with "Intent" -> display the corresponding programme image
        program_id = getIntent().getExtras().getInt("program_id");
        programmeIllustrationImage.setImageResource(illustration_images_array[program_id]);

        //4: Extract the relevant information from "all_programs_detail.xml" file by using XML parser
        try {
            //5: Open program_detail.xml file stored in Assets folder
            InputStream inputStream = getAssets().open("all_programs_detail.xml");

            //6: Use of XML DOM Parser for extracting data
            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance().newInstance();
            DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
            Document document = documentBuilder.parse(inputStream);
            Element element = document.getDocumentElement();
            element.normalize();

            //7: Read all the nodes containing tag "program"
            NodeList nodeList = document.getElementsByTagName("program");

            //8: Loop through all nodes to find the relevant selected program
            boolean program_found = false;
            for (int i = 0; i < nodeList.getLength(); i++) {
                //9: Get "node" in xml file
                Node node = nodeList.item(i);
                Element sub_Element = (Element) node;
                //9: Get the program "title"
                programTitle = sub_Element.getElementsByTagName("title").item(0)
                        .getChildNodes().item(0).getNodeValue();
                programDescription.setText(programTitle);
                //10: Check if the program title is the selected program. If yes, display its detail
                if (programTitle.contains(program_name)) {
                    description = sub_Element.getElementsByTagName("description").item(0)
                            .getChildNodes().item(0).getNodeValue();
                    qualificationLevel = sub_Element.getElementsByTagName("qualification_level").item(0)
                            .getChildNodes().item(0).getNodeValue();
                    duration = sub_Element.getElementsByTagName("duration").item(0)
                            .getChildNodes().item(0).getNodeValue();
                    career = sub_Element.getElementsByTagName("career").item(0)
                            .getChildNodes().item(0).getNodeValue();
                    //11: Change the variable program_found to "true"
                    program_found = true;
                }
            }
            //11: If not found any program in the xml file, assign all variables to "Not found"
            if (!program_found) {
                description = "Not found";
                qualificationLevel = "Not found";
                duration = "Not found";
                career = "Not found";
            }

            //12: Display the extracted program details into the TextView on Layout
            programDescription.setText(description);
            programQualificationLevel.setText(qualificationLevel);
            programDuration.setText(duration);
            programCareer.setText(career);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

PART 7: Add Pop-up menu

(Selft-directed learning)

Add the Popup “Home” Menu to the MainActivity:

Now, add a “**Popup Home Menu**” to your project. The menu contains 3 items:

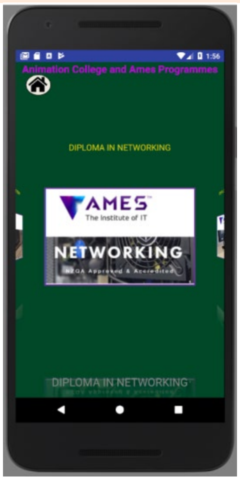
- Item 1: “**About AMES IT**”
- Item 2: “**About Animation College**”
- Item 3: “**Apply now**”
- Item 4: “**Author**”

When use clicks one item, the app opens up a new Activity (layout) to display the relevant information.

Step 1: Copy the “home” menu image (home_menu_image.png) to “drawable” folder:

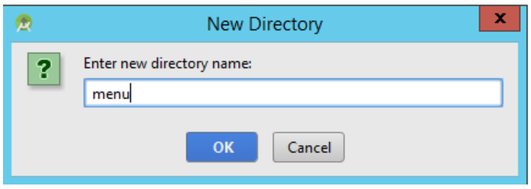
Step 2: Add an ImageButton to activity_main.xml layout:

```
<ImageButton
    android:id="@+id/home_menu"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="20dp"
    android:background="@null"
    android:scaleType="fitCenter"
    android:src="@drawable/home_menu_image" />
```



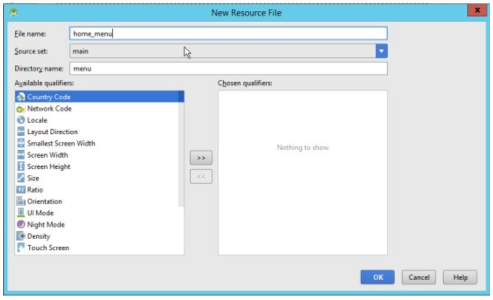
Step 3: Add a new resource file, called home_menu.xml to “menu” folder:

First, add a folder “menu” to the project if it doesn’t exist:



Enter new directory name: menu

Then create a new resource file, called home_menu.xml, inside "menu" folder:



File name: home_menu

Source set: main

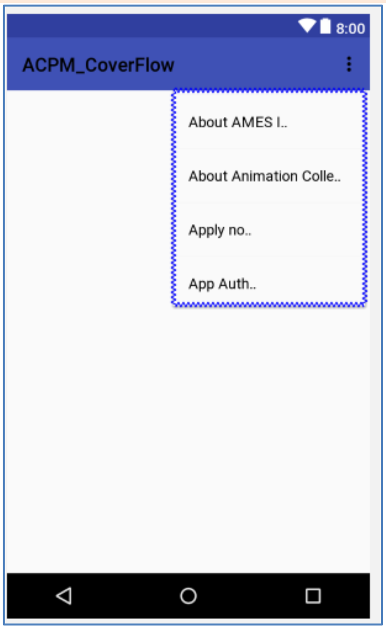
Directory name: menu

Available qualifiers:

Chosen qualifiers:

Finally, edit the home_menu.xml as below:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/aboutAMES"
        android:title="About AMES IT" />

    <item
        android:id="@+id/aboutAC"
        android:title="About Animation College" />

    <item
        android:id="@+id/applyNow"
        android:title="Apply now" />

    <item
        android:id="@+id/author"
        android:title="App Author" />
</menu>
```



Step 4: Inside MainActivity.java, add the below codes:

+ Declare variables:

```
//Popup "Home" menu - 1: Declare variables
private ImageButton homeMenu;
```

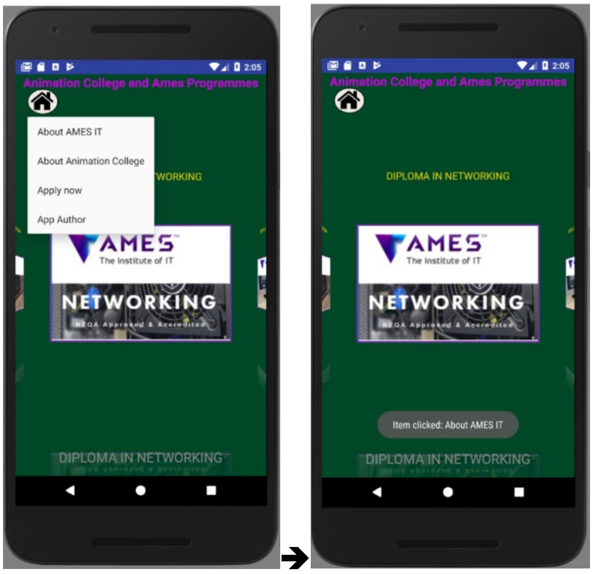
+ Inside onCreate() method, find reference for homeMenu and then set listener for it:

```
/////////////////////////////////////////////////////////////////////
//Popup "Home" menu - 2: Find reference and set listener for homeMenu button
homeMenu = (ImageButton) findViewById(R.id.home_menu);
homeMenu.setVisibility(View.VISIBLE);
homeMenu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //Define Popup Menu for home_menu
        PopupMenu home_menu = new PopupMenu(MainActivity.this, homeMenu);
        //Populate home_menu with items define in "home_menu.xml" file
        home_menu.getMenuInflater().inflate(R.menu.home_menu, home_menu.getMenu());

        //Set on Menu item click listener for home_menu
        home_menu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem menuItem) {
                //
                Toast.makeText(getApplicationContext(), "Item clicked: " + menuItem.getTitle(),
Toast.LENGTH_SHORT).show();
                //
                return false;
            }
        });

        //Show the home_menu
        home_menu.show();
    }
});
```

+ Compile and run the game on AVD:

Run the app, click “home” icon on the top left corner and then click the first item “About AMES IT”.

You’ll see the pop up toast message show “ item clicked: About AMES IT”



Item clicked: About AMES IT

PART 8: Add "About AMES IT" layout

(Selft-directed learning)

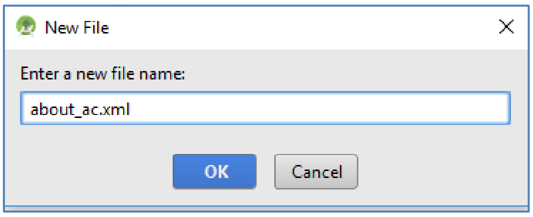
Use "Dialog" to display "About Animation College" layout

Now, when users click one of 3 items in "home" menu:

- Item 1: "**About AMES IT**" → open up a windows to display the content of AMES IT
- Item 2: "**About Animation College**" → open up a windows to display the content of AC
- Item 3: "**Apply now**" → open up a windows to display the online form
- Item 4: "**Author**" → open up a windows to display app developer info

Step 1: Add an xml file (layout) called "how_to_play" & design it

+ Add a new file called "about_ac.xml" to "layout" folder:



New File

Enter a new file name:

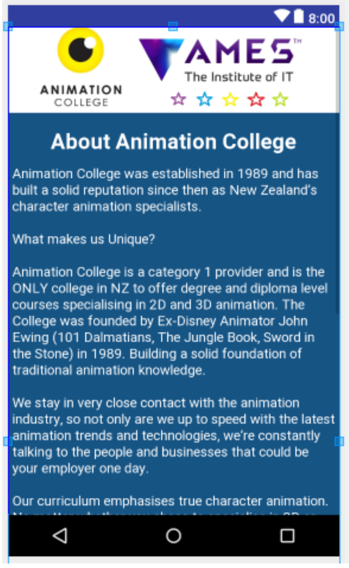
about_ac.xml

+ Inside the file "strings.xml", define the string "about_ac":

```
    <!--About Animation college (AC)-->
    <string name="about_ac">
      Animation College was established in 1989 and has built a solid reputation since then as New Zealand\'s character
animation specialists. \n
\n<b>What makes us Unique?</b>\n
\nAnimation College is a category 1 provider and is the ONLY college in NZ to offer degree and diploma level courses
specialising in 2D and 3D animation.
The College was founded by Ex-Disney Animator John Ewing (101 Dalmatians, The Jungle Book, Sword in the Stone) in 1989.
Building a solid foundation of traditional animation knowledge.  \n
\nWe stay in very close contact with the animation industry, so not only are we up to speed with the latest animation trends
and technologies, we\'re constantly talking to the people and businesses that could be your employer one day.\n
\nOur curriculum emphasises true character animation. No matter whether you chose to specialise in 2D or 3D, the principles
of great animation remain the same. \n
\nAt Animation College you\'ll learn how to create characters that convey real emotions, mimic life, and evoke empathy in
the viewer. You\'ll also learn to tell stories that inspire, influence and entertain. Once learned, you\'ll have those
skills for life and will find that they are transferable to any software, medium, or technology you will encounter in the
industry today or in years to come.\n
\nIt\'s this skill-base that makes our graduates unique and in demand locally and internationally across a wide range of
disciplines, including: animation, illustration, design, gaming, app design, advertising, film and graphic novels or comic
books.\n
\nIf you have a passion for animation, you\'ll love the challenges and opportunities we can offer to expand your creative
potential.
    </string>
```

+ Open and edit about_ac.xml file as below:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#165484"
    android:gravity="center"
    android:orientation="vertical"
    android:scrollbars="vertical">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:scrollbars="vertical">

        <!--Add an ImageView: college logo  -->
        <ImageView
            android:id="@+id/gameLogo"
            android:layout_width="match_parent"
            android:layout_height="100dp"
            android:layout_alignParentTop="true"
            android:layout_marginBottom="10dp"
            android:background="#FFFFFFFF"
            android:contentDescription="collegelogo"
            android:src="@drawable/collegelogo_transparent" />

        <!--Add a TextView: Title-->
        <TextView
            android:id="@+id/title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/gameLogo"
            android:layout_centerHorizontal="true"
            android:gravity="center"
            android:padding="5dp"
            android:text="About Animation College"
            android:textColor="#ffffffff"
            android:textSize="25sp"
            android:textStyle="bold" />

        <!--Add a TextView to display about AC-->
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_below="@+id/title"
            android:layout_centerHorizontal="true"
            android:padding="5dp"
            android:text="@string/about_ac"
            android:textColor="#FFFFFFFF"
            android:textSize="16sp" />
    </RelativeLayout>
</ScrollView>
```



About Animation College

Animation College was established in 1989 and has built a solid reputation since then as New Zealand's character animation specialists.

What makes us Unique?

Animation College is a category 1 provider and is the ONLY college in NZ to offer degree and diploma level courses specialising in 2D and 3D animation. The College was founded by Ex-Disney Animator John Ewing (101 Dalmatians, The Jungle Book, Sword in the Stone) in 1989. Building a solid foundation of traditional animation knowledge.

We stay in very close contact with the animation industry, so not only are we up to speed with the latest animation trends and technologies, we're constantly talking to the people and businesses that could be your employer one day.

Our curriculum emphasises true character animation.

Step 2: Open MainActivity.java, add java codes as below:

+ Declare a Dialog variable:

```
//Dialog to show up how_to_play.xml layout - 1: Declare variable
private Dialog dialog;
```

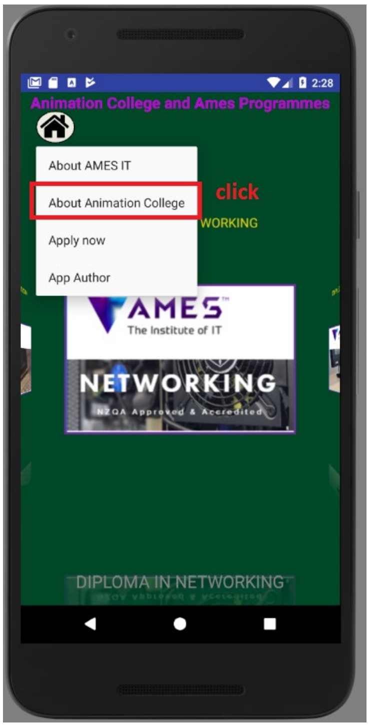
+ Inside the onMenuItemClick() method, add codes to check which menu item has been clicked and then call according function to execute:

```
////////////////////////////////////////////////////////////////////////
//Popup "Home" menu - 2: Find reference and set listener for homeMenu button
homeMenu = (ImageButton) findViewById(R.id.home_menu);
homeMenu.setVisibility(View.VISIBLE);
homeMenu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //Define Popup Menu for home_menu
        PopupMenu home_menu = new PopupMenu(MainActivity.this, homeMenu);
        //Populate home_menu with items define in "home_menu.xml" file
        home_menu.getMenuInflater().inflate(R.menu.home_menu, home_menu.getMenu());

        //Set on Menu item click listener for home_menu
        home_menu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem menuItem) {
                //
                Toast.makeText(getApplicationContext(), "Item clicked: " + menuItem.getTitle(),
Toast.LENGTH_SHORT).show();
                //
                //Dialog to show about_ac.xml layout - 2:
                //check which menu item has been clicked and then call according function to execute
                if (menuItem.getTitle().toString().contains("About Animation College")) {
                    //Item "how to play" has been clicked
                    displayAboutAC();
                }

                return false;
            }
        });

        //Show the home_menu
        home_menu.show();
    }
});
```

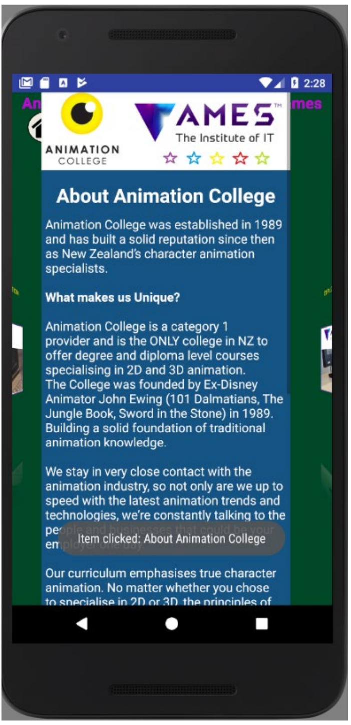
+ Add displayAboutAC() function and edit it as below:

```
//////////////////////////////////////////////////////////////////////////////////////////////
//Dialog to show up about_ac.xml layout - 3: Add and implement displayAboutAC() function
public void displayAboutAC() {
    //Display the about_ac.xml layout
    dialog = new Dialog(MainActivity.this);
    dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
    dialog.getWindow().setBackgroundDrawable(new ColorDrawable(android.graphics.Color.TRANSPARENT));
    dialog.setContentView(R.layout.about_ac);
    dialog.setCanceledOnTouchOutside(true);

    //Determine where to display how_to_play layout on screen:
    Window window = dialog.getWindow();
    WindowManager.LayoutParams wlp = window.getAttributes();
    //
    wlp.gravity = Gravity.LEFT | Gravity.TOP;
    wlp.x = 100; //x position: Here x position's value is pixels from left to right
    wlp.y = 100;// y position:  For y position value is from bottom to top.
    //
    wlp.flags &= ~WindowManager.LayoutParams.FLAG_DIM_BEHIND;
    window.setAttributes(wlp);

    //Show dialog
    dialog.show();
}
```

+ Compile and run the game on AVD:



➔



About Animation College

Animation College was established in 1989 and has built a solid reputation since then as New Zealand's character animation specialists.

What makes us Unique?

Animation College is a category 1 provider and is the ONLY college in NZ to offer degree and diploma level courses specialising in 2D and 3D animation.
The College was founded by Ex-Disney Animator John Ewing (101 Dalmatians, The Jungle Book, Sword in the Stone) in 1989. Building a solid foundation of traditional animation knowledge.

We stay in very close contact with the animation industry, so not only are we up to speed with the latest animation trends and technologies, we're constantly talking to the people and businesses that could be your employer one day.

Item clicked: About Animation College

Our curriculum emphasises true character animation. No matter whether you chose to specialise in 2D or 3D the principles of