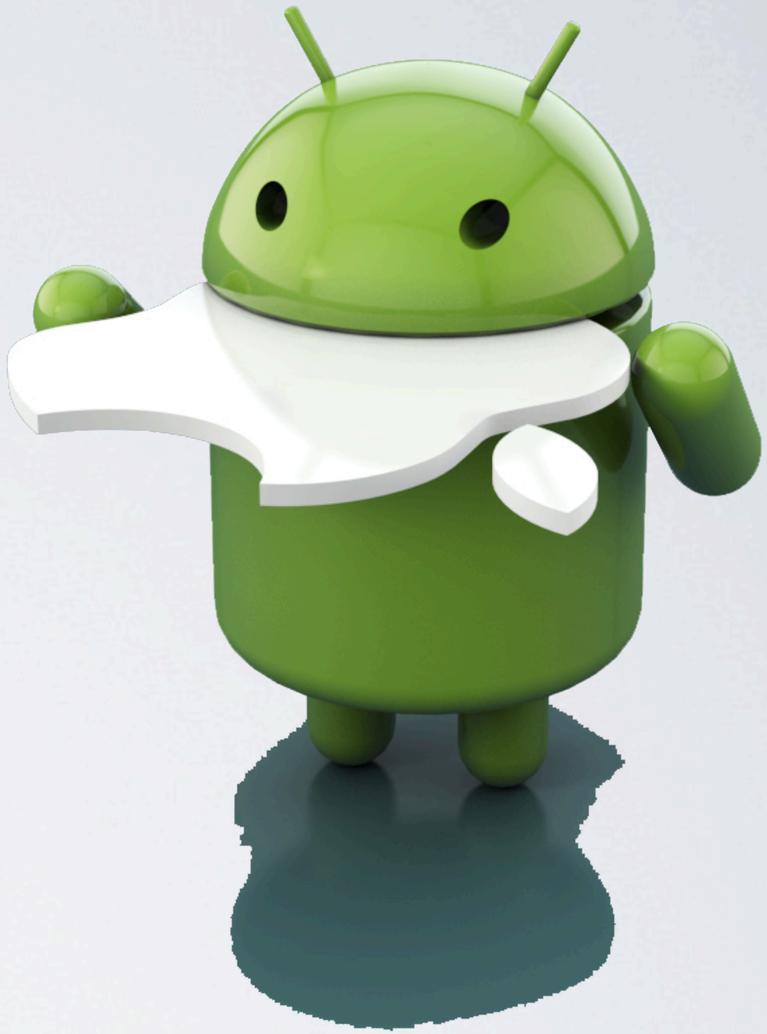

ANDROID

Intro To Android App Creation
Lesson 4

JESSE SCOTT

jscott@langara.bc.ca

MEDIA PLAYBACK

JAVA CLASS - IMPORTS

In our MainActivity.java file, let's add our imports...

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.ImageButton;
import android.media.MediaPlayer;
```

JAVA CLASS - DECLARATIONS

Then, let's add our Class Declarations and Global Variables:

```
// Declarations
private MediaPlayer mp;
private ImageButton button;
```

JAVA CLASS - INSTANTIATIONS

In our onCreate() method, let's instantiate our object & start our player:

```
// Media Player
final MediaPlayer mp = MediaPlayer.create(MainActivity.this, R.raw.sample);
// Button
button = (ImageButton) findViewById(R.id.button);
```

JAVA CLASS - LIFE CYCLE

In our MainActivity Class, let's make the onPause() method to make sure we release the player (i.e. the speakers)

```
@Override
protected void onPause() {
  super.onPause();
  if(mp != null) {
    mp.release();
    mp = null;
  }
}
```

JAVA CLASS - SOUNDS

Inside our res/raw folder, let's add an MP3 called "sample.mp3"

JAVA CLASS - IMAGES

Inside our res/drawable-hdpi folder, let's add a play and a pause icon

JAVA CLASS - MAIN ACTIVITY

In our main_activity.xml file, let's add an Image Button

```
<ImageButton
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="Button"
/>
```

JAVA CLASS - ONCLICK()

Then, let's add our listener for the button:

```
button.setImageResource(R.drawable.img_btn_play);
     button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {
           if(mp.isPlaying()){
              if(mp != null){
                 mp.pause();
                 button.setImageResource(R.drawable.img_btn_play);
              }
           }else{
              if(mp != null){
                 mp.start();
                 button.setImageResource(R.drawable.img_btn_pause);
              }
           }
        }
     });
```

MEDIA RECORDING

JAVA CLASS - IMPORTS

In our MainActivity.java file, let's add our imports...

```
import android.app.Activity;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.media.MediaRecorder;
import java.io.File;
import java.io.IOException;
```

JAVA CLASS - DECLARATIONS

Then, let's add our Class Declarations:

```
// Declarations
private MediaRecorder mRecorder;
```

JAVA CLASS - VARIABLES

Then, let's add our Global Variables:

```
// Variables
public final String TAG = " -- MEDIA_RECORDER -- ";
public final String dirName = "//sdcard//RECORDING";
  public String recFileName =
Environment.getExternalStorageDirectory().getAbsolutePath();
```

ANDROID MANIFEST - PERMISSIONS

In our AndroidManifest.xml file, let's add our permissions...

```
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

JAVA CLASS - MAIN ACTIVITY

In our main_activity.xml file, let's add two Buttons - start & stop

```
<Button
    android:id="@+id/startButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="57dp"
    android:layout_marginTop="74dp"
    android:text="Start"
/>

 <Button
    android:id="@+id/stopButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/startButton"
    android:layout_marginLeft="57dp"
    android:text="Stop"
/>
```

JAVA CLASS - MAKE A DIRECTORY

In our onCreate() method, let's create a custom directory:

```
try{
    File newFile = new File(dirName);
    Log.v(TAG, "dirName is " + dirName);
    newFile.mkdirs();
    if(newFile.exists()) {
      if(newFile.isDirectory()) {
        Log.v(TAG, "Its A Directory...");
      }
    }
    else {
        Log.v(TAG, "Directory Doesn't Exist...");
     }
}
catch(Exception e) {
   e.printStackTrace();
}
```

JAVA CLASS - MAKE A DIRECTORY

In our onCreate() method, let's create our start button:

```
Button startButton = (Button) findViewById(R.id.startButton);
startButton.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    Log.v(TAG, "Clicked Start...");
    startRecording();
  }
});
```

JAVA CLASS - MAKE A DIRECTORY

In our onCreate() method, let's create our stop button:

```
Button stopButton = (Button) findViewById(R.id.stopButton);
stopButton.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    Log.v(TAG, "Clicked Stop...");
    stopRecording();
  }
});
```

JAVA CLASS - LIFE CYCLE

In our MainActivity Class, let's make the onPause() method to make sure we release the recorder (i.e. the microphone)

```
@Override
protected void onPause() {
  super.onPause();
  if(mRecorder != null) {
    mRecorder();
    mRecorder = null;
  }
}
```

JAVA CLASS - START RECORDING

In our MainActivity Class, let's make the startRecording() method

```
public void startRecording() {
    // Set Directory
    recFileName += "//REC//" + "file" + ".3gp";
    // Set MediaRecorder
    mRecorder = new MediaRecorder();
    mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
    mRecorder.setOutputFile(recFileName);
    // ...
```

JAVA CLASS - START RECORDING

In our MainActivity Class, let's make the startRecording() method

```
    // ...
    try {
      mRecorder.prepare();
    }
    catch (IOException e) {
      e.printStackTrace();
    }

    // Record
    mRecorder.start();
}
```

JAVA CLASS - STOP RECORDING

In our MainActivity Class, let's make the stopRecording() method

```
public void stopRecording() {
    mRecorder.stop();
    mRecorder.release();
    mRecorder = null;
}
```