Tutorial 8 – PictureSpinner

1. Go to the following link to access and download the pictures I used in my video tutorial, this way we don’t run into any unforeseen issues.

<https://github.com/amyork/Android_Studio_Tut_8_PictureSpinner/tree/master/app/src/main/res/drawable>

1. Once you have all the files, let’s create a new project named PictureSpinner. Keep everything else generic, and use API 21 or 22. Either will do fine.
2. Delete Hello World!
3. To import the pictures into Android Studio, we need to “copy” all three of the pictures, then inside android studio go to app > res, then right-click on the drawable folder, and select paste. Unfortunately, we cannot simply “drag and drop” images into this folder (at this time). If you chose to get your own pictures for this lesson, make sure every image is named with lowercase letters only. Android studio is very specific about this.
4. Going to our Design View, we need to add two ImageView widgets. Add these however you’d like, then give them id’s of “picture1,” and “picture2.” It may be a little difficult to click on these widgets without any content inside of them using the design view. A better way to select these would be to click on their name in the top right, inside the “component tree.”
5. Click anywhere on the design view that does not contain a widget. We are trying to select the Activity itself, to set a background. Once selected, find “background” in the properties window (bottom right of screen). Click the “…” and find the drawable folder. Select the background image, named “bg” if you have my image files, then click OK. The background will be changed!
6. Next, we need to get our ImageViews to hold images. In the Component Tree, select picture1 and go down to the properties windows. Locate the “src” property, and do the very same as we did for the background, except select the “yellowspin” image. Repeat this for picture2, and use the “arrow” image.
7. The images may appear on top of one another, please separate them to the best of your ability. Leave a little space in between the two images. Once complete, we are done designing.
8. Open up the MainActivity.java file. First, we need to import a few packages:

**import** android.view.View;  
**import** android.widget.ImageView;  
**import** android.view.animation.Animation;  
**import** android.view.animation.RotateAnimation;

1. Within the class, we need to create a few objects. We have two Imageview objects and will be using the RotateAnimation Object:

**private** ImageView **picture1**, **picture2**;  
**private** RotateAnimation **ra**;

1. Now, in the onCreate method, we need to relate our newly created objects to their counterparts on the design view. Add the following lines to the onCreate method:

picture1 = (ImageView)findViewById(R.id.***picture1***);  
**picture2** = (ImageView)findViewById(R.id.***picture2***);

1. The best way to animate these pictures, or say 1000 pictures, would be to create one method to rule them all. In other words, we don’t want 1000 different onClick Listeners, or even 2 in this case. To accomplish this goal, we will create a new function that will be called by each ImageView when clicked. We can accomplish this by adding onClick functionality to the XML file… which we’ll get around to. Create a new function named rotateImage:

**public void** rotateImage(View thisImage)  
{  
// More Code To come!  
}

1. Here’s where the fun begins. We first need to set up our RotateAnimation, which has 6 parameters. Add the following code to the rotateImage function. Do not add the blue text, which is simply a reference for you to understand the parameters.

RotateAnimation ra = **new** RotateAnimation(0, 360, Animation.***RELATIVE\_TO\_SELF***, .5f, Animation.***RELATIVE\_TO\_SELF***, .5f);

(float fromDegrees, float toDegrees, int pivotXType, float pivotXValue, int pivotYType, float pivotYValue)

1. Next we will set up the duration of the rotation, which is in milliseconds. We can also set how many times the rotation will repeat. If you wanted, you can also set the repeat mode! Add the following code to rotateImage function:

ra.setDuration(500); // Takes a half second to rotate 360 degrees  
ra.setRepeatCount(10); // Rotates 10 times before stopping

1. Once we’ve completed “how” we want our animation, we need to simply add code to start the animation. Add the following code to the rotateImage function:

thisImage.startAnimation(ra);

1. Head over to the res folder to find the layout folder. Inside is the activity\_main.xml file. Open this file. Once inside, change over to the Text view (rather than design). At the bottom of each ImageView, add the following line of code:

**Android:onClick="rotateImage"**

1. Now our pictures will both call the function rotateImage when clicked – and this lesson is complete. Load up your emulator and give the program a shot. Please see the linked video for further explanation of the code. Also, note that the emulator isn’t that great when running animations. I have my personal phone running this application at the end of the video where the app runs smoothly.

Video Link: <https://www.youtube.com/watch?v=PJJRt7MWdxo&list=PLFVlCGwfyegYi8G0yxIVlGfjT3xGzCZOz&index=3>

Github Link: <https://github.com/amyork/Android_Studio_Tut_8_PictureSpinner>