Working with ImageView

Prashanth B S[1]

[1]Department of Information Science & Engineering, Nitte Meenakshi Institute of Technology,
Yelahanka - 560064, Bengaluru

1 ImageView

ImageView class is used to display any kind of image resource in the android application either it can be android.graphics.Bitmap or android.graphics.drawable.Drawable. [1]. ImageView is a form of container which holds the image from a particular source, usually from the drawable folder. The sample XML code for adding the ImageView is as shown below,

```
<ImageView
        android:id="@+id/myImg"
        android:layout_width="match_parent"
        android:layout_height="400dp"
        android:src="@drawable/img_name"
         />
```

The properties of the source can be dynamically changed with the help of JAVA controller code as well.

```
ImageView imageView.setImageResource(R.drawable.night); //night.png is the image in
    drawable directory
```

2 Exercise

Create an ImageSwitcher which switches the images from the Drawable folder controlled by the button. Following are the steps,

- Download two android background images from internet, save them as first.png and second.png respectively.
- Copy and Paste the 2 downloaded images into drawable folder under res directory.
- Open the activity_main.xml, Change the Layout to LinearLayout and add an ImageView widget.

```
<ImageView
        android:id="@+id/myImg"
        android:layout_width="match_parent"
        android:layout_height="400dp"
        android:src="@drawable/first" // by default first.png is displayed
        android:layout_margin="20dp"/>
```

[1] https://www.geeksforgeeks.org/imageview-in-android-with-example/

- In the MainActivity.java, Instantiate the ImageView and Button object respectively and equate them with XML objects that forms the view as shown below. Also add an eventListener to the Button as well and override the onclick function.

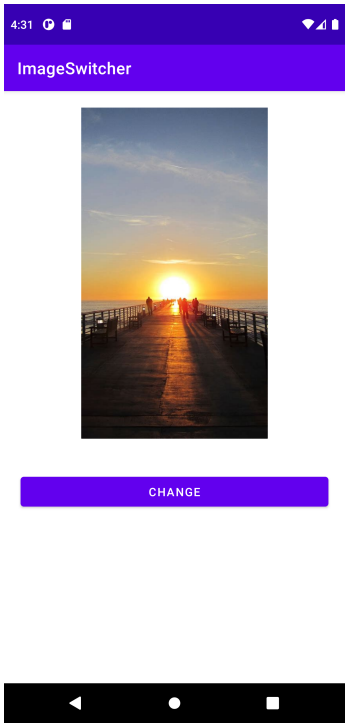
```
ImageView imageView;
Button change ;
int flag = 1 ; // A flag that keep tracks of the image being displayed

// Inside onCreate()
imageView = findViewById(R.id.myImg) ;
change = findViewById(R.id.click);
change.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
                // Code to change the image
    }
});
```

- Inside the onClick function control the image with the help of flag(flag = 1/2, depending on the image being displayed). Use the ImageView.setImgResource(drawable resource) function to change the image from the controller code as shown below,

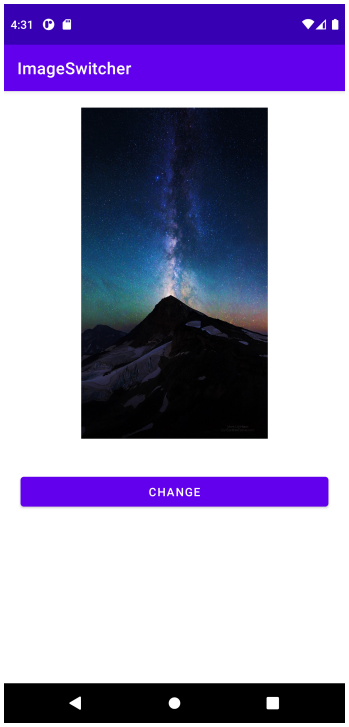
```
change.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(flag==1){
                    imageView.setImageResource(R.drawable.night);
                    flag=2 ;
                }else{
                    imageView.setImageResource(R.drawable.morning);
                    flag=1 ;
                }

            }
        });
```

3 Output

The following 1a shows the App Landing Screen on button click the image gets changed into second as shown in figure 1b.



(a) Imageview displaying the first Image



(b) Imageview displaying the Second Image

Figure 1: ImageFlipper App