# U18CO018

# Shubham Shekhaliya

# Assignment 6

# Develop a Roulette Game for Android

About Roulette game: Roulette is a casino game with a wheel having numbers from 0 to 36. In the Roulette Game, players may choose to place bets on either a single number, various groupings of numbers, red or black colors, whether the number is odd or even, or if the numbers are high (between 19 and 36) or low (between 1 and 18). Then, a croupier spins the wheel in one direction and a little ball in the opposite direction. When the wheel stops, we look at the position of the ball on the sectors of the wheel. After that, the croupier pay players if they won their bets according some rules.

Design of GUI should have the following views:

A TextView to display the result of the wheel’s spin

A Button to spin the wheel

An ImageView to display the wheel which will be represented by image as shown in Figure 1.

An ImageView to display a triangle (shown in Figure 2) pointing to the sector of the wheel where the ball has stopped. So, we won’t use a real ball spinning on the wheel here.



Figure 1: Wheel used for our Roulette Game



Figure 2: Pointer for our Roulette Game

Expected Output: Once the application is launched, you will have the following starting screen:



Click on the spin button, and the wheel will spin. When the wheel stops, your application should show the value of sector pointed by triangle as shown in the following image.



Hint: To rotate wheel, you can use Android Animation API available in the standard SDK, [Butter Knife library](http://jakewharton.github.io/butterknife/) or any other packages depending on implementation language.

Activity\_main.xml

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

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

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    tools:context=".MainActivity">

    <Button

        android:id="@+id/spinBtn"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_alignParentBottom="true"

        android:layout\_centerHorizontal="true"

        android:text="SPIN"

        android:layout\_marginBottom="15dp"

        />

    <TextView

        android:id="@+id/resultTv"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_alignParentTop="true"

        android:layout\_marginTop="15dp"

        android:textSize="32sp"

        android:layout\_centerHorizontal="true"

        android:text=""/>

    <ImageView

        android:id="@+id/wheel"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_centerInParent="true"

        android:layout\_marginLeft="20dp"

        android:layout\_marginRight="20dp"

        android:scaleType="centerInside"

        app:srcCompat="@drawable/wheel"

        android:adjustViewBounds="true" />

    <ImageView

        android:id="@+id/triangle"

        android:layout\_width="25dp"

        android:layout\_height="25dp"

        android:layout\_centerHorizontal="true"

        android:layout\_above="@id/wheel"

        android:layout\_marginBottom="-10dp"

        app:srcCompat="@drawable/triangle"

        />

</RelativeLayout>

MainActivity.java

public class MainActivity extends AppCompatActivity {

    // sectors of our wheel (look at the image to see the sectors)

    private static final String[] sectors = { "32 red", "15 black",

            "19 red", "4 black", "21 red", "2 black", "25 red", "17 black",

"34 red","6 black", "27 red","13 black", "36 red", "11 black",

"30 red", "8 black","23 red", "10 black", "5 red", "24 black",

"16 red", "33 black","1 red", "20 black", "14 red", "31 black",

"9 red", "22 black","18 red", "29 black", "7 red", "28 black",

"12 red", "35 black", "3 red", "26 black", "zero" };

@BindView(R.id.spinBtn)

    Button spinBtn;

    @BindView(R.id.resultTv)

    TextView resultTv;

    @BindView(R.id.wheel)

    ImageView wheel;

    // We create a Random instance to make our wheel spin randomly

    private static final Random RANDOM = new Random();

    private int degree = 0, degreeOld = 0;

    // We have 37 sectors on the wheel, we divide 360 by this value to have angle for each sector

    // we divide by 2 to have a half sector

    private static final float HALF\_SECTOR = 360f / 37f / 2f;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        ButterKnife.bind(this);

    }

    @OnClick(R.id.spinBtn)

    public void spin(View v) {

        degreeOld = degree % 360;

        // we calculate random angle for rotation of our wheel

        degree = RANDOM.nextInt(360) + 720;

        // rotation effect on the center of the wheel

        RotateAnimation rotateAnim = new RotateAnimation(degreeOld, degree,

                RotateAnimation.RELATIVE\_TO\_SELF, 0.5f, RotateAnimation.RELATIVE\_TO\_SELF, 0.5f);

        rotateAnim.setDuration(3600);

        rotateAnim.setFillAfter(true);

        rotateAnim.setInterpolator(new DecelerateInterpolator());

        rotateAnim.setAnimationListener(new Animation.AnimationListener() {

            @Override

            public void onAnimationStart(Animation animation) {

                // we empty the result text view when the animation start

                resultTv.setText("");

            }

            @Override

            public void onAnimationEnd(Animation animation) {

                // we display the correct sector pointed by the triangle at the end of the rotate animation

                resultTv.setText(getSector(360 - (degree % 360)));

            }

            @Override

            public void onAnimationRepeat(Animation animation) {

            }

        });

        // we start the animation

        wheel.startAnimation(rotateAnim);

    }

    private String getSector(int degrees) {

        int i = 0;

        String text = null;

        do {

            // start and end of each sector on the wheel

            float start = HALF\_SECTOR \* (i \* 2 + 1);

            float end = HALF\_SECTOR \* (i \* 2 + 3);

            if (degrees >= start && degrees < end) {

                // degrees is in [start;end[

                // so text is equals to sectors[i];

                text = sectors[i];

            }

            i++;

        } while (text == null  &&  i < sectors.length);

        return text;

    }

}

Output:-

 

 