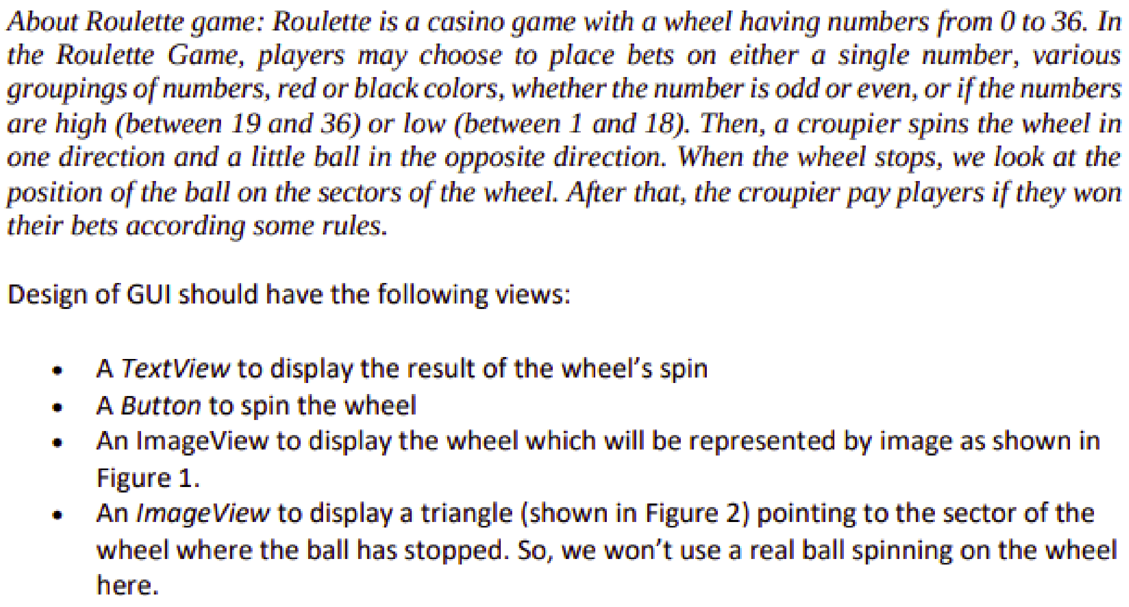
ST LAB 6

SAHIL BONDRE: U18CO021

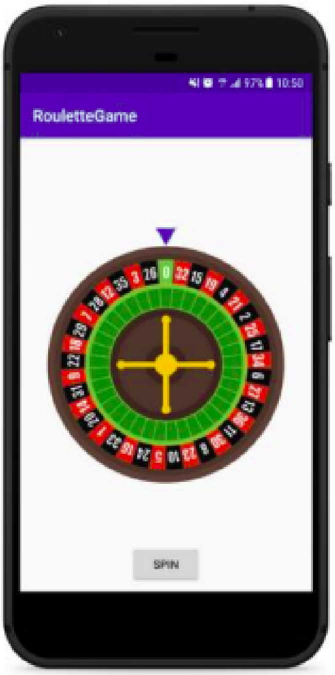


About Roulette game: Roulette is a casino game with a wheel having numbers from 0 to 36. In the Roulette Game, players may choose to place bets on either a single number, various groupings of numbers, red or black colors, whether the number is odd or even, or if the numbers are high (between 19 and 36) 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.

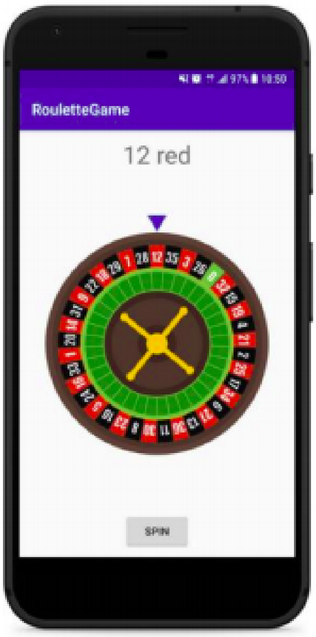
Design of GUI should have the following views:

- 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.

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.



12 red

Hint: To rotate wheel, you can use Android Animation API available in the standard SDK, Butter Knife library or any other packages depending on implementation language.

app build.gradle

```
plugins {
    id 'com.android.application'
}

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.2"

    defaultConfig {
        applicationId "com.example.roulettelab6"
        minSdkVersion 18
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner
"androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles
getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation
'androidx.test.espresso:espresso-core:3.3.0'
    implementation 'com.jakewharton:butterknife:10.0.0'
    annotationProcessor 'com.jakewharton:butterknife-compiler:10.0.0'
}
```

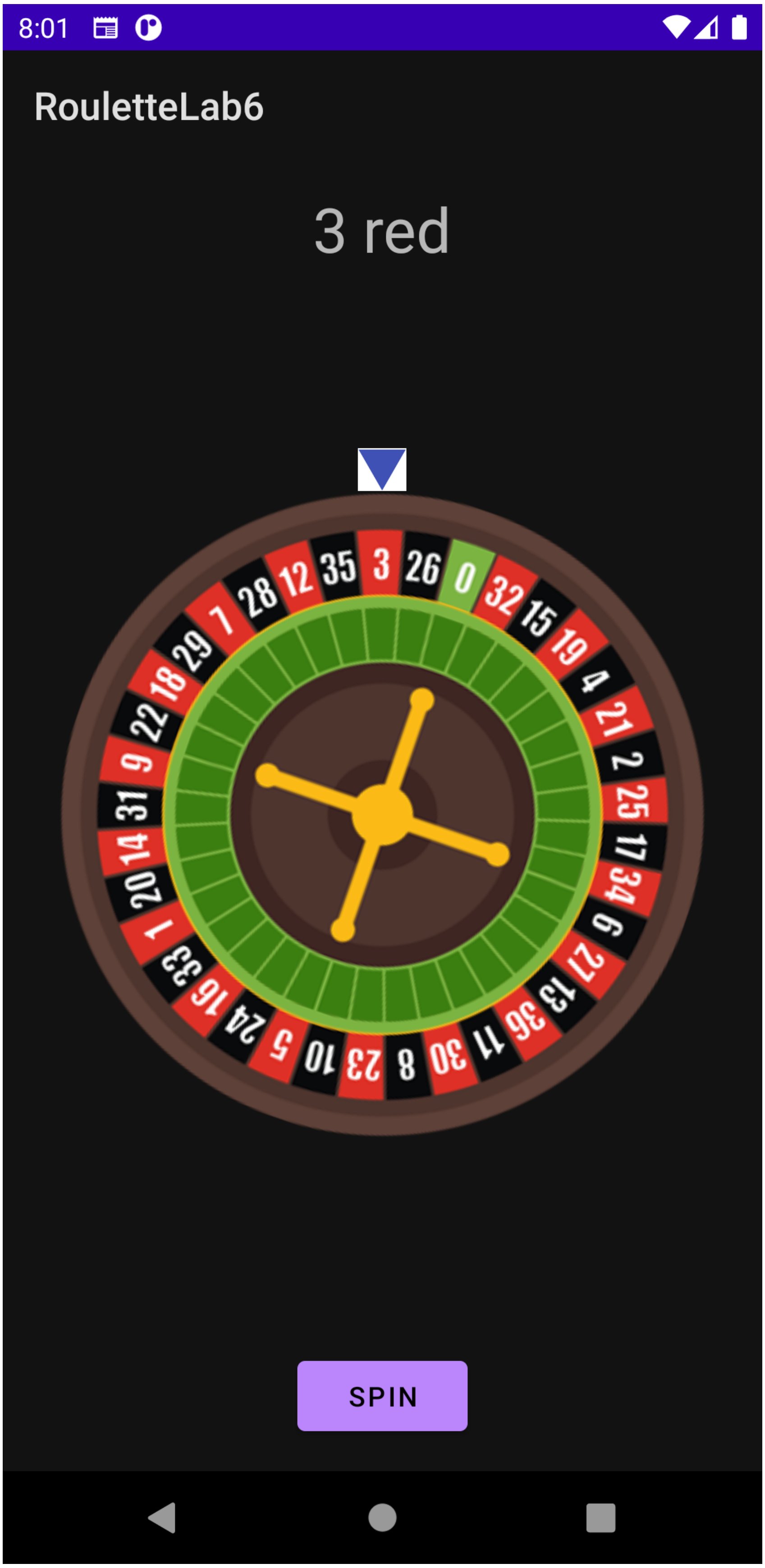
activity_main.xml

```
<?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">

    <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

```
package com.example.roulettelab6;

import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.DecelerateInterpolator;
import android.view.animation.RotateAnimation;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import java.util.Random;

import butterknife.BindView;
import butterknife.ButterKnife;
import butterknife.OnClick;

public class MainActivity extends AppCompatActivity {

    // sectors of wheel
    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;
    // Random instance to make our wheel spin randomly
    private static final Random RANDOM = new Random();
    private int degree = 0, degreeOld = 0;
    // 37 sectors on the wheel, divide 360 by this value to have angle
for each sector
    // 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;
        // calculate random angle for rotation of our wheel
        degree = RANDOM.nextInt(360) + 720;
        // rotation effect on the center of the wheel
        RotateAnimation rotateAnimation = new RotateAnimation(degreeOld,
degree,
                RotateAnimation.RELATIVE_TO_SELF, 0.5f,
RotateAnimation.RELATIVE_TO_SELF, 0.5f);
        rotateAnimation.setDuration(3600);
        rotateAnimation.setFillAfter(true);
        rotateAnimation.setInterpolator(new DecelerateInterpolator());
        rotateAnimation.setAnimationListener(new
Animation.AnimationListener() {
            @Override
            public void onAnimationStart(Animation animation) {
                // empty the result text view when the animation start
                resultTv.setText("");
            }
            @Override
            public void onAnimationEnd(Animation animation) {
                // 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) {

            }
        });

        // start the animation
        wheel.startAnimation(rotateAnimation);
    }

    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;
    }
}
```



RouletteLab6

3 red