

ST LAB 7

U18CO021: SAHIL BONDRE

Extending Assignment 6, here, we are adding certain rules to play *Roulette game* game which are explained below.

Input: *Amount* (in INR) and *Choice*

Amount: You are supposed to enter an amount, let's say INR 500 and you will get the chance to spin the wheel 5 times, i.e., each trial will cost INR 100 and after each trial, INR 100 will be deducted from *Amount*.

Choice: You are supposed to give any one choice out of following four choices during each trial. If after each trial, the output of Roulette game is same as choice given by you for that trial, you will get reward (in INR) as mentioned below:

- | | |
|---|------------------|
| 1. Choice 1: Any odd number between 0 to 36 | Reward: INR 100 |
| 2. Choice 2: Any even number between 0 to 36 | Reward: INR 100 |
| 3. Choice 3: Any Prime number between 0 to 36 | Reward: INR 500 |
| 4. Choice 4: A fixed number between 0 to 36 | Reward: INR 5000 |

Output: For each trial, your program should print following parameters as output:

Trial_number, Remaining_amount, Reward_earned, Trial_left

And after completion of all trials, you are supposed to display

Amount_invested, Total_rewards_earned and ***Net_profit/Net_loss*** earned.

Trial_number indicates how many attempts/trial have been made till now.

Remaining_amount is the Amount left after all trial made till now

Reward_earned means the money you earned after rotation

Trial_left indicates the number of attempt you have.

Amount_invested denotes the initial amount you bet for.

Total_rewards_earned is the total money (in INR) you earned after completion of game/trials.

Net_profit/Net_loss indicates the profit/loss you got.

NOTE: You are not supposed to quit the game without completion.

build.gradle

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

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.2"

    defaultConfig {
        applicationId "com.example.advroulettegame1ab7"
        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:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout

        android:id="@+id/bottomRow"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="15dp">

        <Button
            android:id="@+id/spinOddBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginHorizontal="15dp"
            android:layout_weight="1"
            android:text="Odd" />

        <Button
            android:id="@+id/spinEvenBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginHorizontal="15dp"
            android:layout_weight="1"
            android:text="Even" />

    </LinearLayout>

    <LinearLayout
        android:id="@+id/buttonUpRow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@id/bottomRow"
```

```
android:layout_centerHorizontal="true"  
android:layout_marginBottom="15dp">
```

```
<Button
```

```
    android:id="@+id/spinPrimeBtn"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginHorizontal="15dp"  
  
    android:text="Prime" />
```

```
<EditText
```

```
    android:id="@+id/chooseInput"  
    android:layout_width="50dp"  
    android:layout_height="wrap_content"  
    android:layout_marginHorizontal="15dp"  
    android:layout_marginRight="12px"  
  
    android:inputType="number" />
```

```
<Button
```

```
    android:id="@+id/spinNumberBtn"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginHorizontal="15dp"  
    android:text="Number" />
```

```
</LinearLayout>
```

```
<TextView
```

```
    android:id="@+id/resultTv"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="15dp"  
    android:text=""  
    android:textSize="32sp" />
```

```
<LinearLayout
```

```
    android:id="@+id/topRow"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal">
```

```
<TextView
    android:id="@+id/moneyLeftTv"

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="4dp"
    android:layout_weight="1"
    android:text="Money Left: 0" />
```

```
<TextView
    android:id="@+id/earnedTv"

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="4dp"
    android:layout_weight="1"
    android:text="Earned: 0" />
```

```
<TextView
    android:id="@+id/trialNumberTv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="4dp"
    android:layout_weight="1"
    android:text="Trial: 0" />
```

```
<TextView
    android:id="@+id/trialLeftTv"

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="4dp"
    android:layout_weight="1"
    android:text="Left: 0" />
```

```
</LinearLayout>
```

```
<ImageView
    android:id="@+id/wheel"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/triangle"
```

```
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:adjustViewBounds="true"
    android:scaleType="centerInside"
    app:srcCompat="@drawable/wheel" />
```

<ImageView

```
    android:id="@+id/triangle"
    android:layout_width="25dp"
    android:layout_height="25dp"
    android:layout_below="@+id/resultTv"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="16dp"
    android:layout_marginBottom="-10dp"
    app:srcCompat="@drawable/triangle" />
```

<LinearLayout

```
    android:id="@+id/startRow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/wheel"
    android:layout_centerHorizontal="true">
```

<EditText

```
    android:id="@+id/investInput"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_marginRight="12px"
    android:inputType="number" />
```

<Button

```
    android:id="@+id/startBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start" />
```

</LinearLayout>

<Button

```
    android:id="@+id/finBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/startRow"
    android:layout_centerHorizontal="true"
    android:text="Finish" />
```

```
</RelativeLayout>
```

MainActivity.java

```
package com.example.advroutettegamelab7;

import android.os.Bundle;
import android.os.SystemClock;
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.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import androidx.appcompat.app.AlertDialog;
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"
    };

    // Random instance to make our wheel spin randomly
```

```

    private static final Random RANDOM = new Random();
    // 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;
    private final int ANIMATION_DURATION = 3600;
    @BindView(R.id.spinEvenBtn)
    Button spinEvenBtn;
    @BindView(R.id.spinOddBtn)
    Button spinOddBtn;
    @BindView(R.id.spinPrimeBtn)
    Button spinPrimeButton;
    @BindView(R.id.spinNumberBtn)
    Button spinNumberBtn;
    @BindView(R.id.startBtn)
    Button startBtn;
    @BindView(R.id.finBtn)
    Button finishBtn;
    @BindView(R.id.chooseInput)
    EditText chooseInput;
    @BindView(R.id.investInput)
    EditText investInput;
    @BindView(R.id.resultTv)
    TextView resultTv;
    @BindView(R.id.moneyLeftTv)
    TextView moneyLeftTv;
    @BindView(R.id.earnedTv)
    TextView earnedTv;
    @BindView(R.id.trialNumberTv)
    TextView trialNumberTv;
    @BindView(R.id.trialLeftTv)
    TextView trialLeftTv;
    @BindView(R.id.bottomRow)
    LinearLayout bottomRow;
    @BindView(R.id.buttonUpRow)
    LinearLayout buttonUpRow;
    @BindView(R.id.topRow)
    LinearLayout topRow;
    @BindView(R.id.startRow)
    LinearLayout startRow;
    @BindView(R.id.wheel)
    ImageView wheel;
    private int degree = 0, degreeOld = 0;
    private int moneyLeft = 0;
    private int moneyInvested = 0;
    private int moneyEarned = 0;

```



```

private int trialLeft = 0;
private int trialNum = 0;
private GameState gameState = GameState.START;

private void updateTopRow() {
    trialLeft = moneyLeft / 100;
    moneyLeftTv.setText("Money Left: " + moneyLeft);
    earnedTv.setText("Earned: " + moneyEarned);
    trialLeftTv.setText("Left: " + trialLeft);
    trialNumberTv.setText("Trial: " + trialNum);
}

private void updateUIByState() {
    if (gameState == GameState.START) {
        buttonUpRow.setVisibility(View.INVISIBLE);
        bottomRow.setVisibility(View.INVISIBLE);
        startRow.setVisibility(View.VISIBLE);
        topRow.setVisibility(View.INVISIBLE);
        finishBtn.setVisibility(View.INVISIBLE);
    } else {
        if (moneyLeft >= 100) {
            buttonUpRow.setVisibility(View.VISIBLE);
            bottomRow.setVisibility(View.VISIBLE);
        } else {
            buttonUpRow.setVisibility(View.INVISIBLE);
            bottomRow.setVisibility(View.INVISIBLE);
        }
        startRow.setVisibility(View.INVISIBLE);
        topRow.setVisibility(View.VISIBLE);
        finishBtn.setVisibility(View.VISIBLE);
    }
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ButterKnife.bind(this);
    updateUIByState();
}

@OnClick(R.id.spinEvenBtn)
public void spinEven(View v) {
    int n = spinWheel();
    new Thread(() -> {

```

```

        SystemClock.sleep(ANIMATION_DURATION);
        moneyLeft -= 100;
        if (n % 2 == 0) {
            // even
            moneyEarned += 100;
        }
        runOnUiThread(() -> {

            updateUIByState();
            updateTopRow();

        });
    }).start();
}

```

```

@OnClick(R.id.spinOddBtn)
public void spinOdd(View v) {
    int n = spinWheel();
    new Thread(() -> {
        SystemClock.sleep(ANIMATION_DURATION);
        moneyLeft -= 100;
        if (n % 2 != 0) {
            // even
            moneyEarned += 100;
        }
        runOnUiThread(() -> {

            updateUIByState();
            updateTopRow();

        });
    }).start();
}

```

```

//checks whether an int is prime or not.
private boolean isPrime(int n) {
    for (int i = 2; i < n; i++) {
        if (n % i == 0)
            return false;
    }
    return true;
}

```

```

@OnClick(R.id.spinPrimeBtn)
public void spinPrime(View v) {
    int n = spinWheel();
}

```

```

        new Thread(() -> {
            SystemClock.sleep(ANIMATION_DURATION);
            moneyLeft -= 100;
            if (isPrime(n)) {
                // even
                moneyEarned += 500;
            }
            runOnUiThread(() -> {

                updateUIByState();
                updateTopRow();

            });
        }).start();
    }

    @OnClick(R.id.spinNumberBtn)
    public void spinNumber(View v) {
        int n = spinWheel();
        int chosen = 0;
        if (chooseInput.getText().toString().length() > 0) {
            chosen = Integer.parseInt(chooseInput.getText().toString());
        }
        int finalChosen = chosen;
        new Thread(() -> {
            SystemClock.sleep(ANIMATION_DURATION);
            moneyLeft -= 100;
            if (n == finalChosen) {
                // even
                moneyEarned += 5000;
            }
            runOnUiThread(() -> {

                updateUIByState();
                updateTopRow();

            });
        }).start();
    }

    @OnClick(R.id.finBtn)
    public void finish() {
        AlertDialog alertDialog = new AlertDialog.Builder(this)
//set icon
                .setIcon(android.R.drawable.ic_dialog_alert)
//set title
                .setTitle("Here's you summary")
//set message
    }

```

```

        .setMessage("Invested: " + moneyInvested + "\nRewards
Earned: " + moneyEarned + "\nRevenue: " + (moneyEarned + moneyLeft -
moneyInvested))
//set positive button
        .setPositiveButton("New Game", (dialogInterface, i) -> {
            gameState = GameState.START;
            updateUIByState();
        })
        .show();
    }

    private int spinWheel() {
        ++trialNum;
        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(ANIMATION_DURATION);
        rotateAnimation.setFillAfter(true);
        rotateAnimation.setInterpolator(new DecelerateInterpolator());
        String val = getSector(360 - (degree % 360));
        int sectorNum = Integer.parseInt(val.replaceAll("[\\D]", ""));
        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(val);
            }

            @Override
            public void onAnimationRepeat(Animation animation) {

```

```

        }
    });

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

@OnClick(R.id.startBtn)
public void startGame(View v) {
    moneyLeft = 1500;
    if (investInput.getText().toString().length() > 0) {
        moneyLeft =
Integer.parseInt(investInput.getText().toString());
    }
    moneyInvested = moneyLeft;
    moneyEarned = 0;
    gameState = GameState.RUNNING;
    updateUIByState();
    updateTopRow();
}

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;
}
}

```

GameState.java

```
package com.example.advroutettegame1ab7;  
  
public enum GameState {  
    START,  
    RUNNING,  
}
```

3:51



AdvRouletteGameLab7



START

3:51



AdvRouletteGameLab7

Money Left: 1500

Earned: 0

Trial: 0

Left: 15



FINISH

PRIME

NUMBER

ODD

EVEN

3:52



AdvRouletteGameLab7

Money Left: 1100

Earned: 200

Trial: 4

Left: 11

29 black



FINISH

PRIME

NUMBER

ODD

EVEN

3:52



AdvRouletteGameLab7

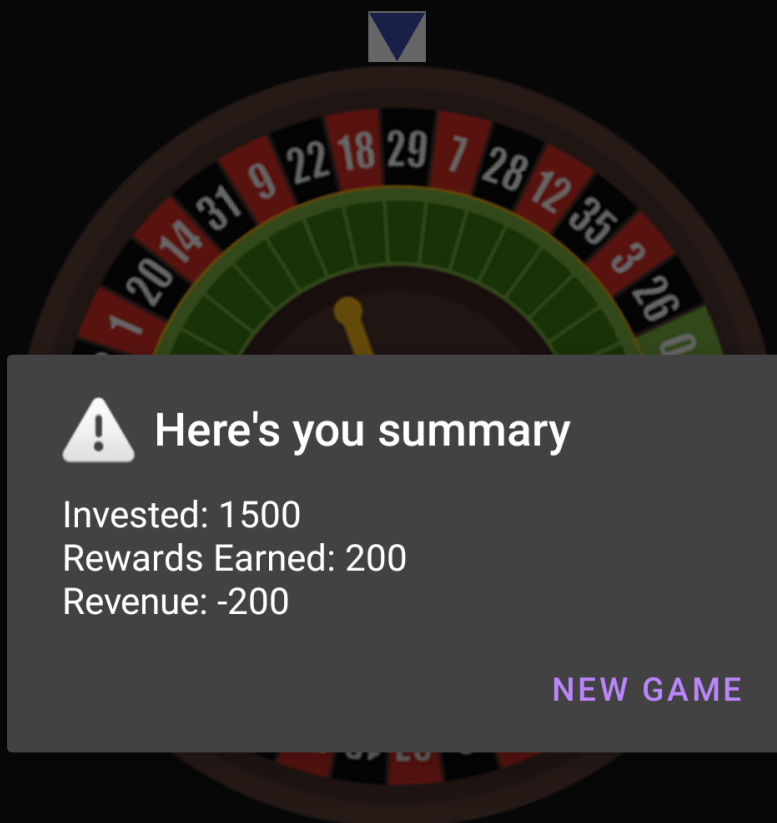
Money Left: 1100

Earned: 200

Trial: 4

Left: 11

29 black



Here's you summary

Invested: 1500

Rewards Earned: 200

Revenue: -200

NEW GAME

FINISH

PRIME

NUMBER

ODD

EVEN