Mobile Application Development Laboratory
106118083 - Rvs Satyanand

1.

Experiment Name: Dice rolling Application

Date: 15-02-2021

Aim: Design an android application using Android Studio for Dice Roll game with the following specifications.
1. Have a button, when clicked - roll the dice and stop at a random number. Visualize the rolling of dice using a set of images with the face of the dice.
2. Have two players; both players will get a chance one after another. The one who scores 25 first will be the winner.
3. Display the winner information and loser information using toasts.

Description of App: The app simulates a dice rolling application which is a multiple player game.

The logic of the app is Once the Application starts Player 1 will start and when player 1 has made his/her move then a random number between (1 to 6) is chosen and that number will be added to the score of Player 1, then player 2 will get a chance to perform his/her move. This way the game will go on till the score of any one player reaches **25** once it reaches game is over and the player who reached or crossed 25 will win. The app uses onClick listeners for the buttons. **The result is shown in the textview and Toast.**

Device Specifications: The app runs on min SDK version of 16 (so anything above API 16 - Android 4.1 - Jelly Bean would run this app which is 99.8% of devices).
Name: Pixel_3

Resolution: 1080 X 2220
API: 30
Target: Android 11.0
hw.lcd.height: 2220
hw.accelerometer: yes
hw.device.manufacturer: Google
hw.lcd.width: 1080
hw.lcd.density: 440
hw.cpu.ncore: 6
hw.sensors.proximity: yes
hw.sensors.orientation: yes
hw.gpu.enabled: yes

Technical Concepts Learnt:

1. Worked with AnimationUtils and loadAnimations library in java to show the rotation animation.
2. Random() function is used to get a pseudo random number.
3. Changing Image Resource in runtime using java.
4. Rotate tag, for example -> to Rotate animation in Clockwise, we need to set android:fromDegrees and android:toDegrees property values and these will define a rotation angle like as shown below.
5. Constraint Layout - Adding Constraints and GuideLines.
6. Instantiating views by findViewById
7. Click Listeners
8. TextView, Button, ConstraintLayout

Source Code:

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
```

```xml
    android:background="#3C1F4E">

    <TextView
        android:id="@+id/tvp1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="P1 = 0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toStartOf="@id/tvp2"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintHorizontal_bias="0.1"
        app:layout_constraintVertical_bias="0.1"
        android:textSize="32dp"
        android:textColor="#48810A"
        android:textStyle="bold"
    />

    <TextView
        android:id="@+id/tvp2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="P2 = 0"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@id/tvp1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintHorizontal_bias="0.9"
        app:layout_constraintVertical_bias="0.1"
        android:textSize="32dp"
        android:textColor="#48810A"
        android:textStyle="bold"
    />

    <ImageView
        android:id="@+id/imgDice"
        android:layout_width="230dp"
        android:layout_height="230dp"
        app:srcCompat="@drawable/fulldice"
        app:layout_constraintTop_toBottomOf="@id/tvp1"
        app:layout_constraintBottom_toTopOf="@id/btnRoll"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:scaleType="fitXY"
        />

    <Button
        android:id="@+id/btnRoll"
        android:layout_width="wrap_content"
```

```xml
        android:layout_height="wrap_content"
        android:backgroundTint="#200E2C"
        android:text="P1 Turn"
        android:textColor="@color/white"
        android:textSize="24dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@id/btnReset"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/imgDice"
        app:layout_constraintVertical_bias="0.9" />

    <Button
        android:id="@+id/btnReset"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/reset"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@id/btnRoll"
        app:layout_constraintVertical_bias="0.9"
        android:textSize="24dp"
        android:textColor="@color/white"
        android:textStyle="bold"
        android:backgroundTint="#200E2C"/>


</androidx.constraintlayout.widget.ConstraintLayout>
```

ActivityMain.java

```java
package com.android.diceroll;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;


import java.util.Random;
```

```java
public class MainActivity extends AppCompatActivity {

    Random rnd = new Random();
    TextView p1ScoreText ,p2ScoreText;
    int p1score=0,p2score=0;
    int turn=1;
    Button rollBtn,resetBtn;
    ImageView diePic;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        diePic = (ImageView)findViewById(R.id.imgDice);
        p1ScoreText=(TextView)findViewById(R.id.tvp1);
        p2ScoreText=(TextView)findViewById(R.id.tvp2);
        rollBtn=(Button)findViewById((R.id.btnRoll));
        rollBtn.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                diceRotate();
            }
        });
        resetBtn=(Button)findViewById(R.id.btnReset);
        resetBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                resetAll();
            }
        });
    }

    private void resetAll() {
        p1score=0;
        p2score=0;
        turn = 1;
        p1ScoreText.setText("P1 = 0");
        p2ScoreText.setText("P2 = 0");
        diePic.setImageResource(R.drawable.fulldice);
        rollBtn.setText("P1 Turn");
        rollBtn.setEnabled(true);


    }
```

```java
    private void diceRotate() {
        int num = rnd.nextInt(6)+1;
        Animation rolling = AnimationUtils.loadAnimation(this,R.anim.rotate);
        diePic.setImageResource(R.drawable.fulldice);
        rolling.setAnimationListener(new Animation.AnimationListener() {
            @Override
            public void onAnimationStart(Animation animation) {
                diePic.setImageResource(R.drawable.fulldice);
            }

            @Override
            public void onAnimationEnd(Animation animation) {
                switch (num){
                    case 1:
                        diePic.setImageResource(R.drawable.die_1);
                        break;
                    case 2:
                        diePic.setImageResource(R.drawable.die_2);
                        break;
                    case 3:
                        diePic.setImageResource(R.drawable.die_3);
                        break;
                    case 4:
                        diePic.setImageResource(R.drawable.die_4);
                        break;
                    case 5:
                        diePic.setImageResource(R.drawable.die_5);
                        break;
                    case 6:
                        diePic.setImageResource(R.drawable.die_6);
                        break;
                }
                if(turn==1)
                {
                    p1score+=num;
                    String p1text = "P1 = "+Integer.toString(p1score);
                    p1ScoreText.setText(p1text);
                    turn*=-1;
                    rollBtn.setText("P2 Turn");
                    if(p1score>=25)
                    {
                        Toast.makeText(MainActivity.this, "Player 1 Won with score
"+p1score, Toast.LENGTH_SHORT).show();
                        rollBtn.setText("Press Reset");
```

```java
                rollBtn.setEnabled(false);
            }
        }
        else if(turn==-1)
        {
            p2score+=num;
            String p2text = "P2 = "+Integer.toString(p2score);
            p2ScoreText.setText(p2text);
            turn*=-1;
            rollBtn.setText("P1 Turn");
            if(p2score>=25)
            {
                Toast.makeText(MainActivity.this, "Player 2 Won with score
"+p2score, Toast.LENGTH_SHORT).show();
                rollBtn.setText("Press Reset");
                rollBtn.setEnabled(false);
            }
        }
    }

    @Override
    public void onAnimationRepeat(Animation animation) {

    }
});
diePic.startAnimation((rolling));


    }
}
```
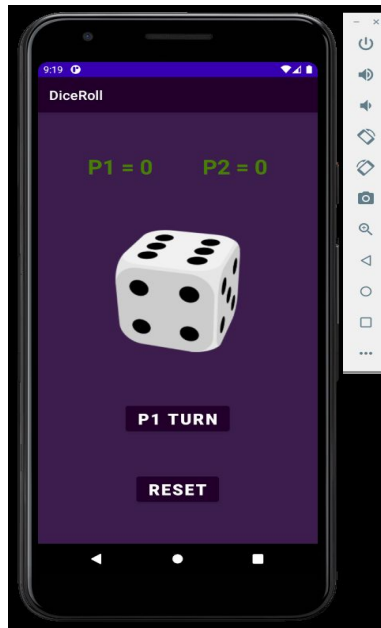
Screenshots:

Do watch the video to see the animation in action. Link ->
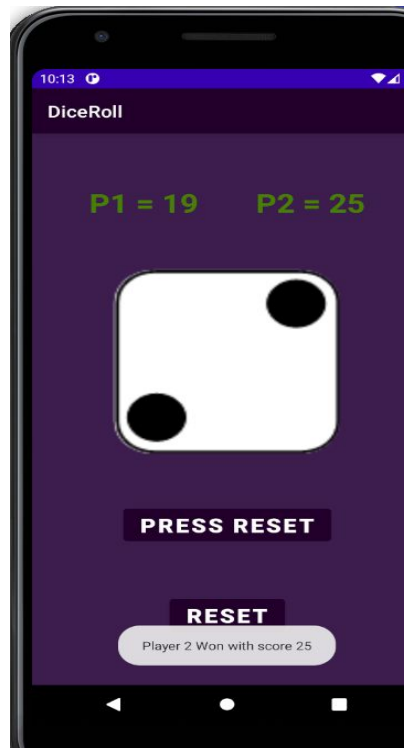**https://drive.google.com/file/d/1iDS8xW4Jju77RTEcbLNG005utMtEIBel/view ?usp=sharing**
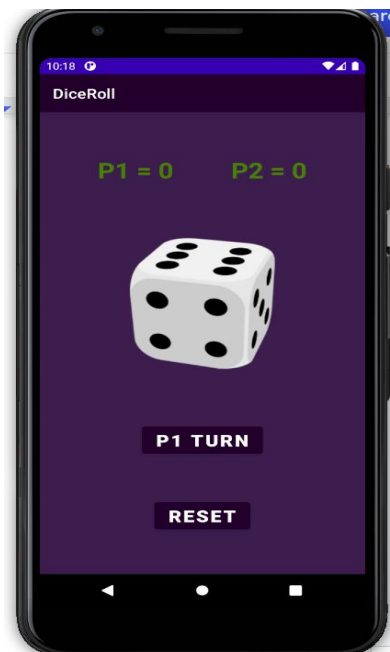
i) Start:

ii) When Player 1 presses the button and random number comes out to be 4 then player 1 score gets incremented by 4 and now its player 2 turns.

iii) When the player wins the game and then there is a toast message showing the result.

iv) After pressing the **reset** button the game is re-initialized.

Outcomes:
The tasks given were accomplished without any bugs/crashes.