
App 포트폴리오

안드로이드를 이용한 포커게임 시뮬레이션

이 름	김민성
제작기간	2016/10/10 ~ 2016/10/24

1

개요

셔플링 할 때, 카드가 중복되어 나와선 안되도록 프로그래밍 하였고, 5장의 카드를 카드 그림(참고 자료)를 사용하여, 이미지 뷰를 통해서 출력하였습니다. 또한, 카드 규칙에 맞게 족보를 출력하도록 하였고, 현재까지의 족보별 출현횟수를 바탕으로, 현 횟수와 확률을 계산한 후, 다른 액티비티에서 출력되도록 하였습니다. 마지막으로, 시뮬레이션을 출력할 액티비티를 따로 두고, 카드뽑기를 10,000번 실행한 시뮬레이션 통계자료를 출력하도록 하였습니다.

2

소스코드

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="s20120209.ce.kumoh.ac.kr.mypoker">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:largeHeap="true"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:screenOrientation="landscape"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".StatisticsActivity"
            android:label="통계 보기"
            android:screenOrientation="landscape"></activity>
        <activity
            android:name=".SimulationActivity"
            android:label="시뮬레이션 보기"
            android:screenOrientation="landscape"></activity>
    </application>
```

</manifest>

MainActivity.java

```
package s20120209.ce.kumoh.ac.kr.mypoker;

import android.content.Intent;
import android.content.res.Resources;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    int[] number = new int[5];
    int[] CardNum = new int[5];          //뽑힌 카드 숫자
    int[] CardPattern = new int[5];      //뽑힌 카드 무늬
    ImageView[] iv = new ImageView[5];
    TextView tv;
    BitmapDrawable[] CardImage = new BitmapDrawable[52]; //포커 카드 이미지

    //통계 자료 보기 데이터
    int[] statistics_data = new int[13]; //족보별 출현횟수 카운팅

    //시뮬레이션 자료 보기 데이터
    boolean simulation_check = false;
    int[] simulation_data = new int[13];

    /**
     * 포커 Card 족보
     * 모양
     * 0x10 : ♣
     * 0x20 : ♥
     * 0x30 : ♦
     * 0x40 : ♠
     * 숫자
     * 0x02 ~ 0x09 : 2~9
     * 0x0A : 10
     * 0x0B : J 11
     */
}
```

```

* 0x0C : Q 12
* 0x0D : K 13
* 0x0E : A 14
*/
int[] CardName = {
    0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D, 0x1E,
    0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x2A, 0x2B, 0x2C, 0x2D, 0x2E,
    0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, 0x3A, 0x3B, 0x3C, 0x3D, 0x3E,
    0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4D, 0x4E,
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    iv[0] = (ImageView) findViewById(R.id.image1);
    iv[1] = (ImageView) findViewById(R.id.image2);
    iv[2] = (ImageView) findViewById(R.id.image3);
    iv[3] = (ImageView) findViewById(R.id.image4);
    iv[4] = (ImageView) findViewById(R.id.image5);

    tv = (TextView) findViewById(R.id.textView);

    /**
     * 포커 카드 이미지 불러오기
     */
    Resources res = getResources();
    CardImage[0] = (BitmapDrawable) res.getDrawable(R.drawable.c2); //0x12
    CardImage[1] = (BitmapDrawable) res.getDrawable(R.drawable.c3); //0x13
    .... 생략 ....
    CardImage[50] = (BitmapDrawable) res.getDrawable(R.drawable.sk); //0x4D
    CardImage[51] = (BitmapDrawable) res.getDrawable(R.drawable.sa); //0x4E

    .... 생략 (Tool Bar , Snack Bar 관련 함수) ....

    /**
     * 카드 뽑기 함수
     *
     * @param view
     */
    public void onRandom(View view) {

        for (int i = 0; i < number.length; i++) {
            number[i] = (int) (Math.random() * 51) + 0;    //포커카드 뽑기 (0 ~ 51)
        }
    }

```

```

        //현재까지 뽑은 번호에 대해서, 중복체크
        for (int j = 0; j < i; j++) {
            if (number[j] == number[i]) {
                i--;    //중복된 수가있다면, 카운트를 감소시키고, 다시 뽑기
                break;
            }
        }
    }
}

checkCardRule(number); //카드 족보 출력

//카드를 이미지뷰에 출력
for (int i = 0; i < number.length; i++) {
    iv[i].setImageDrawable(CardImage[number[i]]);
}
}

/**
 * 카드 족보 체크 함수
 */
public void checkCardRule(int[] number) {

    boolean OnePair = false;    //원페어 : 같은숫자 2장
    boolean TwoPair = false;    //투페어 : 원페어 2쌍
    boolean Treeple = false;    //트리플 : 같은숫자 3장
    boolean Straight = false;   //스트레이트 : 무늬와 관계없이, 5장의 연속된 숫자
    boolean BackStraight = false; //백 스트레이트 : 무늬와 관계없이 A,2,3,4,5
    boolean Mountain = false;   //마운틴 : 무늬와 관계없이 A,K,Q,J,10
    boolean Flush = false;      //플러시 : 숫자와 관계없이 같은 무늬 5장
    boolean FullHouse = false;  //풀하우스 : 같은숫자3장, 같은숫자2장
    boolean FourCard = false;   //포카드 : 같은숫자 4개
    boolean StraightFlush = false; //스트레이트 플러시 : 5장 무늬가 모두 같고, 연속된 숫자
    boolean BackStraightFlush = false; //백 스트레이트 플러시 : 5장 모두 같은무늬, A,2,3,4,5
    boolean RoyalStraightFlush = false; //로얄 스트레이트 플러시 : 5장모두 같은무늬, A,K,Q,J,10

    int[] card = new int[5];
    //카드 숫자로 변환(Mapping 작업)
    for (int i = 0; i < number.length; i++) {
        card[i] = CardName[number[i]];
    }

    //card의 무늬, 숫자를 분리
    for (int i = 0; i < number.length; i++) {
        CardNum[i] = card[i] & 0x0F;    //카드 숫자
        CardPattern[i] = card[i] & 0xF0; //카드 무늬
    }
}

```

```

//버블 정렬(오름차순 정렬)
for (int i = 0; i < CardNum.length - 1; i++) {
    for (int j = 0; j < CardNum.length - i - 1; j++) {

        if (CardNum[j] > CardNum[j + 1]) {
            int temp = CardNum[j];
            CardNum[j] = CardNum[j + 1];
            CardNum[j + 1] = temp;
            int temp2 = number[j];
            number[j] = number[j + 1];
            number[j + 1] = temp2;
        }
        if (CardPattern[j] > CardPattern[j + 1]) {
            int temp = CardPattern[j];
            CardPattern[j] = CardPattern[j + 1];
            CardPattern[j + 1] = temp;
        }
    }
}

//카드번호 출현횟수 카운팅
int[] CardNumCount = new int[15];
for (int i = 0; i < CardNum.length; i++)
    CardNumCount[CardNum[i]]++;
for (int i = 0; i < CardNumCount.length; i++) {
    if (CardNumCount[i] >= 4)
        FourCard = true;
    if (CardNumCount[i] >= 3)
        Treeple = true;
    if (CardNumCount[i] >= 2) {
        if (OnePair == true)
            TwoPair = true;
        OnePair = true;
    }
}

//로얄 스트레이트 플러시 체크
if (NumIsStraight(CardNum) && PatternIsEqual(CardPattern) &&
    CardNum[0] == 0x0A && CardNum[4] == 0x0E) { // 10,J,Q,K,A
    RoyalStraightFlush = true;
    tv.setText("로얄 스트레이트 플러시");
    if (simulation_check)
        simulation_data[0]++;
    else

```

```

        statistics_data[0]++;
    return;
}

//백 스트레이트 플러시 체크
if (NumIsStraight(CardNum) && PatternIsEqual(CardPattern) &&
    CardNum[0] == 0x02 && CardNum[4] == 0x0E && CardNum[2] == 0x04) { // 2,3,4,5,A
    BackStraightFlush = true;
    tv.setText("백스트레이트 플러시");

    if (simulation_check)
        simulation_data[1]++;
    else
        statistics_data[1]++;
    return;
}

//스트레이트 플러시 체크
if (NumIsStraight(CardNum) && PatternIsEqual(CardPattern)) {
    StraightFlush = true;
    tv.setText("스트레이트 플러시");

    if (simulation_check)
        simulation_data[2]++;
    else
        statistics_data[2]++;
    return;
}

//포카드 체크
if (FourCard) {
    tv.setText("포카드");
    FourCard = true;

    if (simulation_check)
        simulation_data[3]++;
    else
        statistics_data[3]++;
    return;
}

//풀하우스 체크
if (Treeple && TwoPair) {
    FullHouse = true;
    tv.setText("풀하우스");

    if (simulation_check)

```

```

        simulation_data[4]++;
    else
        statistics_data[4]++;
    return;
}

//플러시 체크
if (PatternIsEqual(CardPattern)) {
    Flush = true;
    tv.setText("플러시");

    if (simulation_check)
        simulation_data[5]++;
    else
        statistics_data[5]++;
    return;
}

//마운틴 체크
if (NumIsStraight(CardNum) && CardNum[0] == 0x0A && CardNum[4] == 0x0E) { // 10,J,Q,K,A
    Mountain = true;
    tv.setText("백 스트레이트");

    if (simulation_check)
        simulation_data[6]++;
    else
        statistics_data[6]++;
    return;
}

//백스트레이트 체크
if (NumIsStraight(CardNum) && CardNum[0] == 0x02 && CardNum[4] == 0x0E && CardNum[2] == 0x04) {
    BackStraight = true;
    tv.setText("백 스트레이트");

    if (simulation_check)
        simulation_data[7]++;
    else
        statistics_data[7]++;
    return;
}

//스트레이트 체크
if (NumIsStraight(CardNum)) {
    Straight = true;
    tv.setText("스트레이트");

    if (simulation_check)

```



```

        simulation_data[8]++;
    else
        statistics_data[8]++;
    return;

}

//트리플 체크
if (Treeple) {

    tv.setText("트리플");

    if (simulation_check)
        simulation_data[9]++;
    else
        statistics_data[9]++;
    return;
}

//투페어 체크
if (TwoPair) {
    tv.setText("투페어");

    if (simulation_check)
        simulation_data[10]++;
    else
        statistics_data[10]++;
    return;
}

//원페어 체크,
if (OnePair) {
    tv.setText("원페어");

    if (simulation_check)
        simulation_data[11]++;
    else
        statistics_data[11]++;
    return;
}

//노 페어(No Pair)
tv.setText("노페어");

if (simulation_check)
    simulation_data[12]++;
else
    statistics_data[12]++;
return;

```

```

}

//같은 무늬가 있는지 체크
public boolean PatternIsEqual(int[] CardPattern) {
    for (int i = 0; i < CardPattern.length - 1; i++) {
        if (CardPattern[i] != CardPattern[i + 1])
            return false;
    }

    return true;
}

//스트레이트가 있는지 체크
public boolean NumIsStraight(int[] CardNum) {

    boolean AceChance = false;
    if (CardNum[0] == 0x02 && CardNum[4] == 0x0E) //중간에 에이스가 낀 경우,
        AceChance = true;

    for (int i = 0; i < CardNum.length - 1; i++) {
        if (CardNum[i] + 1 != CardNum[i + 1]) {
            if (AceChance) {
                AceChance = false;
                continue;
            } else
                return false;
        }
    }

    return true;
}

/**
 * 통계 보기 버튼 함수
 */
public void onStatistics(View view) {
    Intent intent = new Intent(this, StatisticsActivity.class);
    intent.putExtra("statistics_data", statistics_data);
    startActivity(intent);
}

/**
 * 시뮬레이션 버튼 함수
 */
public void onSimulation(View view) {

    //시뮬레이션 데이터 초기화

```

```

        for (int i = 0; i < simulation_data.length; i++)
            simulation_data[i] = 0;

        simulation_check = true; //시뮬레이션 시작
        //10000번 실행
        for (int count = 0; count < 10000; count++) {
            for (int i = 0; i < number.length; i++) {
                number[i] = (int) (Math.random() * 51) + 0;    //포커카드 뽑기 (0 ~ 51)

                //현재까지 뽑은 번호에 대해서, 중복체크
                for (int j = 0; j < i; j++) {
                    if (number[j] == number[i]) {
                        i--;    //중복된 수가있다면, 카운트를 감소시키고, 다시 뽑기
                        break;
                    }
                }
            }
        }

        checkCardRule(number); //카드 족보 카운팅
    }

    Intent intent = new Intent(this, SimulationActivity.class);
    intent.putExtra("simulation_data", simulation_data);
    simulation_check = false; //시뮬레이션 종료
    startActivity(intent);
}
}

```

content_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:background="@drawable/poker_board"
    android:orientation="vertical"

    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="s20120209.ce.kumoh.ac.kr.mypoker.MainActivity"
    tools:showIn="@layout/activity_main">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```

        android:orientation="horizontal">

        <ImageView
            android:id="@+id/image1"
            android:layout_width="100sp"
            android:layout_height="150sp"
            android:layout_margin="5sp"
            android:src="@drawable/c2"
            android:text="Hello World!" />

        <ImageView
            android:id="@+id/image2"
            android:layout_width="100sp"
            android:layout_height="150sp"
            android:layout_margin="5sp"
            android:src="@drawable/c3"
            android:text="Hello World!" />

        <ImageView
            android:id="@+id/image3"
            android:layout_width="100sp"
            android:layout_height="150sp"
            android:layout_margin="5sp"
            android:src="@drawable/c4"
            android:text="Hello World!" />

        <ImageView
            android:id="@+id/image4"
            android:layout_width="100sp"
            android:layout_height="150sp"
            android:layout_margin="5sp"
            android:src="@drawable/c5"
            android:text="Hello World!" />

        <ImageView
            android:id="@+id/image5"
            android:layout_width="100sp"
            android:layout_height="150sp"
            android:layout_margin="5sp"
            android:src="@drawable/c6"
            android:text="Hello World!" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:onClick="onRandom"
    android:text="카드 뽑기"
    android:textSize="45dp" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:background="#ffaaddee"
    android:padding="16dp"
    android:text="죽보 출력"
    android:textColor="#ff00"
    android:textSize="25dp" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:onClick="onStatistics"
    android:text="통계 보기"
    android:textSize="20dp" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:text="시뮬레이션"
    android:onClick="onSimulation"
    android:textSize="20dp" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

StatisticsActivity.java

```
package s20120209.ce.kumoh.ac.kr.mypoker;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;

import java.text.DecimalFormat;

/**
 * Created by Administrator on 2016-09-29.
 */
public class StatisticsActivity extends AppCompatActivity {

    int [] statistics_data;

    TextView [] mTextView_Statistics = new TextView[26];
    int all_statistics_data = 0;
    TextView mTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_statistics);

        Intent intent = getIntent(); //MainActivity 데이터를 받음.
        statistics_data = intent.getIntArrayExtra("statistics_data");
        mTextView = (TextView)findViewById(R.id.textView);
        mTextView_Statistics[0] = (TextView)findViewById(R.id.RoyalStraightFlush2);
        mTextView_Statistics[1] = (TextView)findViewById(R.id.RoyalStraightFlush3);
        mTextView_Statistics[2] = (TextView)findViewById(R.id.BackStraightFlush2);

        ... 생략 ...

        mTextView_Statistics[24] = (TextView)findViewById(R.id.NoPair2);
        mTextView_Statistics[25] = (TextView)findViewById(R.id.NoPair3);

        for(int i=0;i<statistics_data.length;i++) {
            all_statistics_data += statistics_data[i];
        }

        for(int i=0;i<statistics_data.length;i++) {
            double rate = ((double) statistics_data[i] / (double) all_statistics_data) * 100;
            String DigitFormat = "0.####";
```

```

        DecimalFormat form = new DecimalFormat(DigitFormat);
        String Executionrate = form.format(rate);
        Executionrate = Executionrate.concat("%");

        //텍스트뷰에 통계자료 출력
        mTextView_Statistics[(i*2)].setText(Executionrate);
        mTextView_Statistics[(i*2) + 1].setText(statistics_data[i] + "회");
    }

    // mTextView.setText(sum + "");
}
}

```

activity_statistics.xml

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout 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"
    android:background="@drawable/poker_board"
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="s20120209.ce.kumoh.ac.kr.mypoker.StatisticsActivity">

    <TableRow>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_margin="5dp"
            android:background="#ffaaddee"
            android:padding="3dp"
            android:text="축보"
            android:textColor="#ff00"
            android:textSize="20dp"

            />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_margin="5dp"
            android:background="#ffaaddee"

```

```
        android:padding="3dp"
        android:text="확률"
        android:textColor="#ff00"
        android:textSize="20dp"
```

```
    />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="5dp"
    android:background="#ffaaddee"
    android:padding="3dp"
    android:text="출현 횟수"
    android:textColor="#ff00"
    android:textSize="20dp"
```

```
    />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="5dp"
    android:background="#ffaaddee"
    android:padding="3dp"
    android:text="족보"
    android:textColor="#ff00"
    android:textSize="20dp"
```

```
    />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="5dp"
    android:background="#ffaaddee"
    android:padding="3dp"
    android:text="확률"
    android:textColor="#ff00"
    android:textSize="20dp"
```

```
    />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```



```

        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="출현 횟수"
        android:textColor="#ff00"
        android:textSize="20dp"

        />
    </TableRow>

```

```

<TableRow>

```

```

    <TextView
        android:id="@+id/NoPair1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="노페어"
        android:textColor="#ff00"
        android:textSize="14dp "

        ....생략 ....

```

```

    <TextView
        android:id="@+id/RoyalStraightFlush3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="312회"
        android:textColor="#ff00"
        android:textSize="12dp"

        />

```

```

    </TableRow>
</TableLayout>

```

SimulationActivity.java

```

package s20120209.ce.kumoh.ac.kr.mypoker;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;

import java.text.DecimalFormat;

public class SimulationActivity extends AppCompatActivity {

    int[] simulation_data;
    TextView[] mTextView_Statistics_Simulation = new TextView[26];
    TextView mTextView;

    int[] number = new int[5];

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_simulation);

        Intent intent = getIntent();    //MainActivity 데이터를 받음.
        simulation_data = intent.getIntArrayExtra("simulation_data");
        mTextView = (TextView) findViewById(R.id.textView);
        mTextView_Statistics_Simulation[0] = (TextView) findViewById(R.id.SimRoyalStraightFlush2);
        mTextView_Statistics_Simulation[1] = (TextView) findViewById(R.id.SimRoyalStraightFlush3);
        mTextView_Statistics_Simulation[2] = (TextView) findViewById(R.id.SimBackStraightFlush2);
        ... 생략 ...
        mTextView_Statistics_Simulation[24] = (TextView) findViewById(R.id.SimNoPair2);
        mTextView_Statistics_Simulation[25] = (TextView) findViewById(R.id.SimNoPair3);

        for(int i=0;i<simulation_data.length;i++) {
            double rate = ((double) simulation_data[i] / (double) 10000) * 100;
            String DigitFormat = "0.####";
            DecimalFormat form = new DecimalFormat(DigitFormat);
            String Executionrate = form.format(rate);
            Executionrate = Executionrate.concat("%");

            //텍스트뷰에 통계자료 출력
            mTextView_Statistics_Simulation[(i*2)].setText(Executionrate);
            mTextView_Statistics_Simulation[(i*2) + 1].setText(simulation_data[i] + "회");
        }
    }
}

```

activity_simulation.xml

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout 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"
    android:background="@drawable/poker_board"
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="s20120209.ce.kumoh.ac.kr.mypoker.SimulationActivity">

```

```

<TableRow>

```

```

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="죽보"
        android:textColor="#ff00"
        android:textSize="20dp"

```

```

    />

```

```

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="확률"
        android:textColor="#ff00"
        android:textSize="20dp"

```

```

    />

```

```

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="출현 횟수"
        android:textColor="#ff00"

```

```

        android:textSize="20dp"

    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="죽보"
        android:textColor="#ff00"
        android:textSize="20dp"

    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="확률"
        android:textColor="#ff00"
        android:textSize="20dp"

    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="출현 횟수"
        android:textColor="#ff00"
        android:textSize="20dp"

    />
</TableRow>

<TableRow>

    <TextView
        android:id="@+id/SimNoPair1"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="노페어"
        android:textColor="#ff00"
        android:textSize="14dp"

```

.... 생략

<TextView

```

        android:id="@+id/SimRoyalStraightFlush3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#ffaaddee"
        android:padding="3dp"
        android:text="312회"
        android:textColor="#ff00"
        android:textSize="12dp"

```

/>

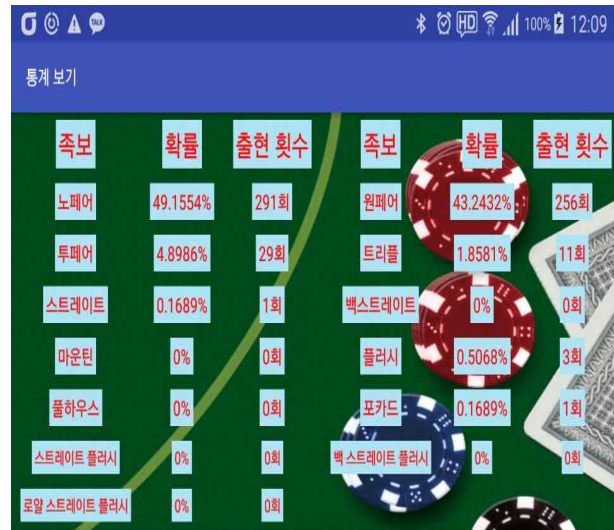
</TableRow>

</TableLayout>

3 실행 결과







4 과제 수행시 어려웠던 점과 해결방법

이전 과제에서, 로또 번호를 생성하는 것을 진행하였었기 때문에, 카드를 셔플링 하여 이미지뷰에 출력시키는 것은 어렵지 않았습니다. 다만, 카드의 족보를 출력하기위해선, 카드를 무늬와 숫자를 구별하기위한 ID 같은 것이 필요하게되었고, 이를 구현하기 위해서, 무늬의 경우엔 16진수의 2번째 자리숫자를 통해서, 구별하고 숫자의 경우엔 16진수의 1번째 자리수를 이용하였습니다. 즉, 카드에 따른 고유 ID가 생기므로, 랜덤으로 분배된 수에 이 ID를 매핑 하여 해결하였습니다. 또한, 카드의 족보를 구별하는 소스코드를 작성하는 과정에서 어려움을 겪었는데, 스트레이트를 구별하는 과정에서 어려움을 겪었습니다. 스트레이트의 경우엔, 연속된 숫자가 5번 나오면 스트레이트로 보는데 이때, 2,3,4,5,6,7,8,9,10,J,Q,K,A 다음에 숫자가 다시 2로 이어지기 때문에 이를 해결하기위해서, 운영체제 시간에 배웠던 Victim Process 선정 알고리즘 중에 있던, Second Chance Algorithm을 사용하였습니다. 예를 들어, 스트레이트로 연결되는 수 중에서, A ~ 2가 연결된 수가 나온다면, 오름차순으로 정렬하면 2가 맨앞에, A가 맨뒤에 위치하게되는데 이것이 나오면 스트레이트 규칙이 만족한다는 것으로 보고 기회를 한번 주도록하여 스트레이트를 검사하도록 하였습니다. 또한, 통계 자료에 대한 데이터와 시뮬레이션 자료에 대한 데이터 카운팅을 구분할 필요가 있었는데, 이를 해결하기위해서, simulation_check(MainActivity.java 참고) 변수를 사용하여, 시뮬레이션을 실행하게되면, 이 변수가 true가 되게 되고, 족보의 출현횟수를 카운팅 하는과정에서 시뮬레이션 족보의 출현

횃수 데이터로 카운팅되게됩니다. 만약, 이 변수가 false라면, 통계자료를 위한 족보의 출현횃수 데이터로 카운팅하게됩니다. 시뮬레이션이 다 끝나게되면, simulation_check 변수를 다시 false로 만들어주게됩니다. 또한, 시뮬레이션을 다시 시작할 경우엔, 이전에 있던 데이터를 모두 0으로 해주어서, 다음 시뮬레이션에 영향을 주지않도록 하였습니다.