实验编号：2 **四川师大Android高级开发实验报告 2020**年 **3**月 **20**日

**计算机科学学院** 级 3 班 实验名称： 随机纸牌

姓名： 吴文娇 学号： 2017110543 指导老师： 樊相奎 实验成绩:\_\_\_\_\_

**实验 二 随机纸牌**

1. 实验目的及要求

目的：

(1) 完善第一个实验的内容;

(2) 通过引入 Model，实现随机纸牌翻牌功能;

要求：

**(**1) 认真填写实验报告，要求附加部分运行界面和主要代码;

(2) 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决;

1. 实验内容

实现 Card、PlayingCard、Deck、PlayingDeck 四个类;

a) Card 为抽象类;

b) PlayingCard 从 Card 派生，实现标准纸牌;

c) Deck 为抽象类;

d) PlayingDeck 从 Deck 派生，实现 52 张牌的初始化;

1. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）

(1). Card、PlayingCard、Deck、PlayingDeck 四个类：

a).Card.java

**package** com.example.fairy.myapplication.model;  
  
**import** java.util.ArrayList;  
**public abstract class** Card {**private boolean chosen**;  
 **private boolean matched**;  
 **public abstract** String getContents();  
 **public boolean** isChosen() { **return chosen**; }  
 **public boolean** isMatched() { **return matched**; }  
 *//定义了两个方法：为了访问private变量（直接return，同get属性）* **public int** match(Card[] otherCards){  
 *// 作用不大* **int** score = 0;  
 **for**(Card card : otherCards){  
 **if** (card.getContents().equals(**this**)){  
 score = 1;  
 }  
 }  
 **return** score;  
 }  
  
}

b).PlayingCard.java

**package** com.example.fairy.myapplication.model;  
  
**import** java.util.Arrays;  
**import** java.util.ArrayList;  
**public class** PlayingCard **extends** Card{  
 *//由Card派生的一个子类：实现54张牌* **private** String **suit**;*//花色* **private int rank**;*//大小* **public** String getSuit() {  
 **return suit**;  
 }  
 *//同Card里的isChosen()和isMatched()：为了访问private变量* **public void** setSuit(String aSuit){  
 **if** (Arrays.*asList*(PlayingCard.*validSuits*()).contains(aSuit)){  
 *//判断花色是不是黑红樱方（Arrays：java的一个类 ）  
 //asList里的字符串数组转化成Array对象，来判断这样的一个字符是不是在数组里面  
 //validSuits()是静态方法* **suit** = aSuit;  
 }  
 }  
 *// 为了给private属性赋值：set  
 // 定义为private变量属性而不是public是为了避免别人想赋值就赋值，自己无法检查的情况* **public static** String[] validSuits(){**return new** String[]{**"♠︎"**,**"♣︎"**,**"♥︎"**,**"♦︎"**};}  
 *// 静态方法：可以用类名调用：上面判断花色的时候调用：PlayingCard.validSuits()  
 // String[]数组是无法用来判断字符在不在这个数组里面的，所以上面需要把它转化成Array数组才可以用来判断* **public int** getRank(){ **return rank**; }  
 **public void** setRank(**int** aRank){  
 **if**(aRank <= PlayingCard.*maxRank*()){  
 **rank** = aRank;  
 }  
 }  
 *//检查了上界：小于等于13* **private static** String[] rankStrings(){  
 **return new** String[]{**"?"**,**"A"**,**"2"**,**"3"**,**"4"**,**"5"**,**"6"**,**"7"**,**"8"**,**"9"**,**"10"**,**"J"**,**"Q"**,**"K"**};  
 *//?:填充符（为了让A到K是1-13）* }  
  
 **public static int** maxRank(){**return** PlayingCard.*rankStrings*().**length** - 1; }  
 @Override  
 **public int** match(Card[] otherCards){  
 *// 判断牌是否相同：即大小相同：4分，花色相同：1分，其他：0分* **int** score = 0;  
 **if** (otherCards.**length** == 1){  
 PlayingCard otherCard = (PlayingCard) otherCards[0];  
 **if**(otherCard.getRank()==**this**.getRank()){  
 score = 4;  
 }**else if** (otherCard.getSuit().equals(**this**.getSuit())){  
 score = 1;  
 }  
 }  
 **return** score;  
 }  
 @Override  
 **public** String getContents(){ **return** PlayingCard.*rankStrings*()[**rank**] + **suit**;}  
 *//返回牌面*}

c).Deck.java

**package** com.example.fairy.myapplication.model;  
  
**import** java.util.ArrayList;  
**import** java.util.Random;  
**public abstract class** Deck {  
 *//一摞牌* **private** ArrayList<Card> **cards** = **new** ArrayList<>();  
 *//表示可以存多少张牌* **private** Random **r** = **new** Random();  
 *// 随机数* **public void** addCard(Card card,**boolean** atTop){  
 **if** (atTop)  
 {  
 **cards**.add(0,card);  
 }**else** {  
 **cards**.add(card);  
 }  
 }  
 **public void** addCard(Card card){addCard(card,**false**);}  
 *//加牌* **public** Card drawRandomCard(){  
 Card randomCard = **null**;  
 **if**(**cards**.size()>0){  
 randomCard = **cards**.remove(**r**.nextInt(**cards**.size()));  
 }  
 **return** randomCard;  
 }  
 *//抽牌*}

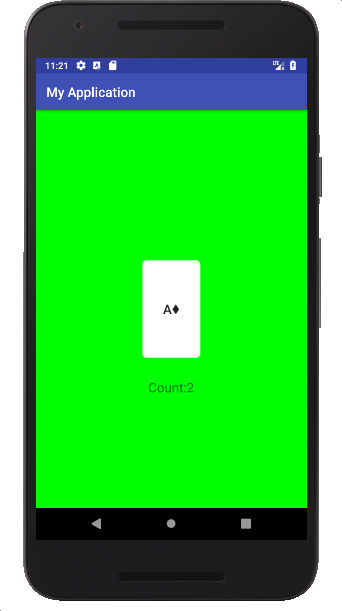
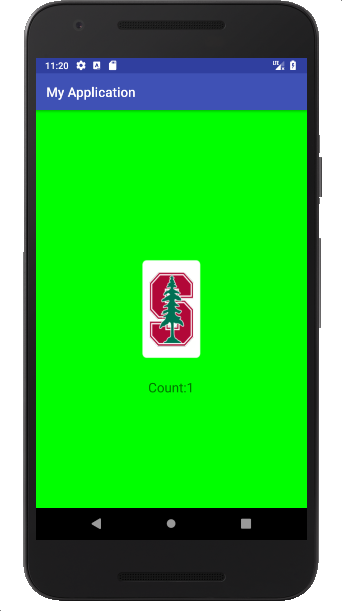
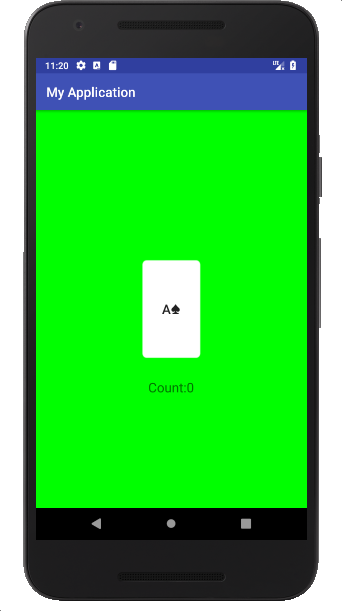
d).PlayingDeck,java

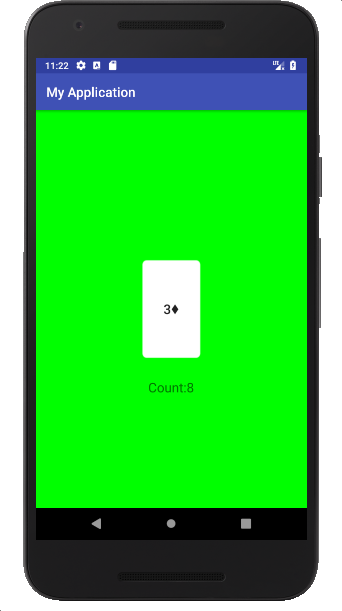
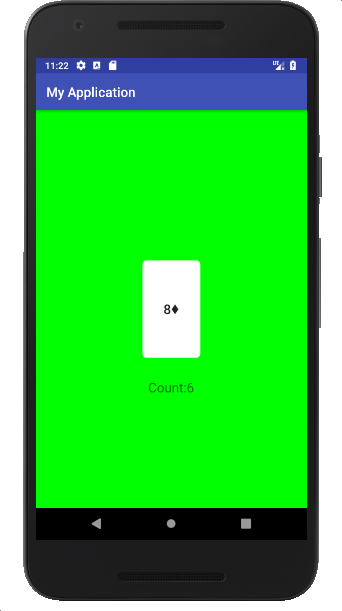
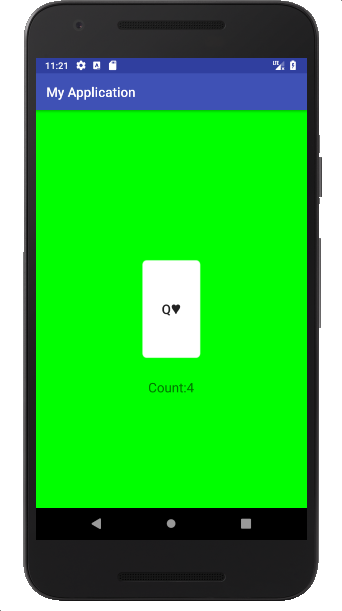
**package** com.example.fairy.myapplication.model;**public class** PlayingDeck **extends** Deck{  
 *//两个for循环构造出52张牌* **public** PlayingDeck(){  
 **super**();  
 **for** (String suit :  
 PlayingCard.*validSuits*()){  
 **for** (**int** rank = 1 ; rank <= PlayingCard.*maxRank*(); rank++){  
 PlayingCard card = **new** PlayingCard();  
 card.setRank(rank);  
 card.setSuit(suit);  
 addCard(card);  
 }  
 }  
 }  
}

(2) 在主 Activity 中实现对 PlayingDeck 的利用

**package** com.example.fairy.myapplication;  
  
**import** android.media.AudioManager;  
**import** android.media.SoundPool;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
  
**import** com.example.fairy.myapplication.model.Card;  
**import** com.example.fairy.myapplication.model.PlayingDeck;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **int count** = 0;  
 TextView **textView**;  
 Button **button**;  
  
 *//声音池* SoundPool **sp** = **new** SoundPool(5, AudioManager.***STREAM\_MUSIC***,0);  
 **int SoundID**;  
  
 PlayingDeck **playingDeck**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***card\_layout***);  
  
 **SoundID** = **sp**.load(**this**,R.raw.***meow***,0);  
 **textView** = (TextView)findViewById(R.id.***count\_tv***);  
 **button** = (Button)findViewById(R.id.***card\_button***);  
  
 **playingDeck** = **new** PlayingDeck();  
  
 **button**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Button button = (Button)view;  
 **if**(button.getText().toString().equals(**""**)){  
 button.setBackgroundResource(R.drawable.***blankcard***);  
 Card card = **playingDeck**.drawRandomCard();  
 **if**(card != **null**)button.setText(card.getContents());  
 }**else**{  
 button.setBackgroundResource(R.drawable.stanfordtree);  
 button.setText(**""**);  
 }  
 count++;  
 textView.setText(**"Count:"**+count);  
 sp.play(SoundID,1.0f,1.0f,1,0,1.0f);  
 }  
 });  
 }  
}

1. 实验结果的分析与评价（该部分如不够填写，请另加附页）





注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。