**项目编号：**  **16712051**

**东 南 大 学**

**大学生创新训练计划项目结题报告**

**项目名称：**  基于android的UNO纸牌游戏app

**项目级别：** 院级

**项目负责人：** 李鑫 学号：（ 71114410 ）

**电话：**  18652082356

**E-mail：**  [18652082356@163.com](mailto:18652082356@163.com)

**项目参加者：** 方舟 （学号： 71114412 ）

关龙涛 （学号： 71114411 ）

**项目指导教师1：** 朱夏 **（所属院系：** 计算机科学与工程学院 **）**

**电话：**  15195758308

**E-mail：**  zhuxia@seu.edu.cn

**项目迄止时间：**  2016 年 1 月-- 2017 年 6 月

**东 南 大 学 教 务 处**

一、项目实施情况

a) 研究目标

过去的纸牌基本只限定在扑克牌范围内，当然扑克牌玩法的多样性、娱乐性深受人们的喜爱。近年来，桌游在国内发展迅速，人们尤其是年轻人更愿意去尝试新的纸牌游戏，UNO便是其中之一。UNO起源于欧洲，如今已经成为了许多人聚会时喜爱的纸牌游戏，多样性以及利于活跃气氛的玩法使得聚会的气氛活跃融洽。但是，一些人完全没有接触过这一款游戏，在聚会时若有这项活动会显得不自然，和其他人会有隔阂。在这样需求下，我们希望用一款安卓平台下的app让玩家尽可能地在最短的时间内熟悉游戏规则并且学习到一些UNO游戏的小技巧。在之后和朋友们玩的时候就会迅速融入到他们中去，说不定还能大展身手。

综上所述，我们希望用户迅速熟悉UNO这款纸牌游戏的游戏规则并且在熟悉规则的基础上了解一些游戏技巧。

b) 研究过程

项目初期，我们对app的架构进行了设计，对于整个项目的类结构进行了设计，结果如下：

1. Color——颜色类，是一个枚举类型，用于展示颜色信息，包括基本颜色（红、黄、蓝、率）和黑色（万能牌）

2. Type——类型类，是一个枚举类型，用于展示卡牌类型信息，包括颜色牌、功能牌等等。

3. SelectActivity——选择模式，选择游戏模式的activity

4. MainActivity——选择等级，选择电脑玩家的AI等级的activity

5. GameActivity——进行游戏，游戏进行时的activity

6. Card——卡牌类，表示每个uno卡牌的基本信息，包括颜色、数字、类型、ID、绑定的bitmap等。

7. CardGroup——卡牌集类，有两个方面：一方面是存储一次游戏的牌组信息，当游戏开始需要发牌和玩家需要抓牌时提供从已洗牌的牌组中提取卡牌；另一方面是作为每个玩家中的成员变量存储玩家的当前手牌

8. Player——玩家类，存储玩家的信息和方法，包括手牌、出牌控制、抓牌控制等。

9. GameController——游戏控制类，以游戏规则为基准，根据游戏具体的情况，控制游戏如何进行，包括顺序进行，方向反转，跳过等等。

10. UNOView——游戏绘画类，整个游戏的关键部分，在游戏界面上画出相应的卡牌、颜色变换的闪烁效果、出牌抓牌时的移动效果，根据游戏的情景不同画出相应的界面，保证游戏正常有效地进行。

11. EventAction——事件监听类，监听游戏的按下时间，与UNOView类进行，提供处理事件的接口，减少UNOView类的压力，减少冗余，具有良好的封装性。

12. Common——通用类，提供一些通用的静态方法，包括位置调整、游戏AI处理等。

项目进入开发阶段。我们针对UNO卡牌的特征进行了Card、CardGroup、Player的实现。随后进行游戏画面的绘制。收集并制作游戏所需要的图片资源，包括游戏背景、游戏卡牌、出牌按钮、推荐按钮、牌组图片、方向指示图片等等。游戏与其他应用不同，它需要频繁的画面变换，这种变换是基于游戏规则的，而非安卓控件的简单变换，所以我们抛弃了布局文件，选择自定义SurfaceView的方式，采用多线程编程的方式，在SurfaceCreated方法中创建两个线程：游戏线程和绘图线程。绘图线程通过画图函数不断在界面中绘制游戏图像，游戏线程根据游戏的具体情况改变游戏状态与游戏元素等等，这也相应的改变了界面的绘制效果。两个线程同时运行才能保证游戏的正常进行。

绘图线程以Surface自身为参数创建线程，重写SurfaceView类中的run方法进行游戏画面的绘制，监听游戏是否运行以及游戏画面是否需要重绘的变量，当满足条件的时候调用OnDraw方法，在这个方法中画出游戏所需的各个元素，包括卡牌、用于指示当前玩家的手指、各个按钮、方向指示、计时器等等。

在Surface创建的时候将我们所需的图片资源加载到游戏中进行初始化图片的工作，随后初始化玩家、牌组并进行洗牌工作，完成游戏的准备工作。之后启动游戏线程，模拟现实生活中游戏开始抓牌的方式，以一次一张的方式将每人的起始7张牌发到每个人的手中，每发一张牌便对画面进行更新，展示出发牌的具体效果。紧接着选择牌库第一张牌作为起始牌，选择一个玩家作为第一个出牌的玩家，在循环中对当前的玩家编号进行监听。当当前玩家为电脑时，在计时器为0时，调用电脑的AI接口，针对电脑AI的选择进行游戏界面和局面的变换，当当前玩家为app使用者时，等待并监听玩家的点击事件，uno纸牌游戏监听的事件只有按下事件一种，通过获取玩家按下的坐标在EventAction中进行相关的处理，包括返回点击的是哪张牌、抓牌、出牌、推荐等等。当玩家打出万能牌或王牌时，弹出选择游戏的界面，玩家选择后，我们运用为画笔设定透明度变化的方式以渐变的方式显示卡牌颜色的变化。在每一次出牌时监测每个玩家的手牌，当有一个玩家手牌数为零时，启动gameWin方法进行积分，得出本局游戏的排名情况。

我们根据网络上收集到的UNO游戏技巧以及自身以往的经验设计了“聪明”电脑的AI选择，首先遍历手牌，根据游戏场景，若没有牌可以出便返回null表示出牌，若可以出则根据不同的场景设定进行不同的出牌选择以及颜色选择。例如：当下架只剩下一张时，选择出功能牌来给下家制造麻烦，若是万能牌还要尽可能避开下家的颜色；当自己有少量牌时，进行最优的留牌选择等等。

项目后期，我们针对不同的游戏场景对于app进行了测试工作，解决了我们所发的漏洞，保证app正常有效地运行。

c) 研究成果

项目基本完成了项目初期设定的目标，能够在玩家完全不熟悉UNO纸牌游戏规则的情况下，让玩家迅速熟悉游戏规则、适应游戏节奏、甚至学习一些有用的游戏技巧。玩家在打开app后可以选择普通模式和训练模式。训练模式中，玩家可以看到其他电脑玩家的手牌，获取到他们出牌的选择，更有利于玩家学习游戏玩法并训练自己的能力。进入游戏后，轮到玩家选择时，玩家可以通过点击纸牌后出牌按钮是否为彩色判断选择的卡牌是否可以打出。如果玩家不知道该如何出牌，可以点击推荐图标，我们会根据游戏规则为替玩家选择该出哪张牌。若没有牌可以出，具有闪烁效果的牌组图标会提醒玩家要从牌组中抓取一张牌，若这张牌满足出牌条件根据UNO的游戏的规则也会直接打出。在游戏结束时，玩家会看到计分及排名信息，了解到计分规则。通过明牌、推荐、计分等功能，玩家可以迅速学习到游戏的规则，在下一次和亲朋好友聚会时就可以大展身手了。

d) 研究心得

开发的规范很重要，包括命名的规范、代码风格规范、接口规范等等。规范的开发方法会使项目更有效率，避免在无谓的事情上花费大量的时间。

安卓项目在内存方面需要考虑得更多，不同机器的内存不同，要花一定的时间解决内存溢出的问题。

在多线程的开发上，我的收获很多。多线程引发的问题往往不是每次都发生的，是不定时发生的bug。我们应多考虑到同步的问题，而不是依赖于代码在形式上面的顺序，考虑到这一点才能对多线程引发的问题找到比较好的解决办法。

项目开发过程中让我领悟到了对于基本原理的理解的重要性。在遇到问题时，我们有时选择在网络上搜寻其他人对于类似问题的解答，问题虽然可以得到顺利解决，但是我们不知道为什么这么做。而当之后再遇到类似问题的时候，便手足无措。归根结底是没有深刻理解底层基本的原理，底层的原理才是解释为什么这么做的关键。所以在之后的学习和工作的过程中，我们不能不求甚解，而要做到知其然和知其所以然，才能提高效率，少走弯路。

团队协作也很重要，一个融洽的团队合作氛围和良好的沟通方法也是项目的关键。

二、项目创新和特色

app商店已经有了一些与UNO有关的纸牌游戏，但是它们大多数进行联机游戏的，单机的游戏并不会受到网络环境的束缚。除此之外，它们与我们项目的目标不同，我们的目标用户是刚刚接触这个游戏的人群，在对刚刚接触这个游戏的友好度要高于其他app。市场上并没有一款这样面对新玩家的应用，我们为新玩家提供了很好的服务，玩家可以让其他电脑玩家明牌，在不知道如何出牌时我们可以像玩家推荐出牌。在游戏开始之前，玩家可以指定三个电脑玩家分别是“一般”还是“聪明”，根据电脑玩家之间的对比，玩家也可以从中迅速获取UNO的游戏技巧，在将来和朋友聚会时的时候就可以使用了。

三、项目成果

软件：1份

研究或实验设计方案：1份

结题报告：1份

心得体会：3份

电子展板：1份

四、研究体会与心得

开发的规范很重要，包括命名的规范、代码风格规范、接口规范等等。规范的开发方法会使项目更有效率，避免在无谓的事情上花费大量的时间。

安卓项目在内存方面需要考虑得更多，不同机器的内存不同，要花一定的时间解决内存溢出的问题。

在多线程的开发上，我的收获很多。多线程引发的问题往往不是每次都发生的，是不定时发生的bug。我们应多考虑到同步的问题，而不是依赖于代码在形式上面的顺序，考虑到这一点才能对多线程引发的问题找到比较好的解决办法。

项目开发过程中让我领悟到了对于基本原理的理解的重要性。在遇到问题时，我们有时选择在网络上搜寻其他人对于类似问题的解答，问题虽然可以得到顺利解决，但是我们不知道为什么这么做。而当之后再遇到类似问题的时候，便手足无措。归根结底是没有深刻理解底层基本的原理，底层的原理才是解释为什么这么做的关键。所以在之后的学习和工作的过程中，我们不能不求甚解，而要做到知其然和知其所以然，才能提高效率，少走弯路。

团队协作也很重要，一个融洽的团队合作氛围和良好的沟通方法也是项目的关键。

五、经费报销情况

无