**结题报告**

**Final report**

**问题描述：**

我们小组选择的是实践类，意向是写出一个2048的游戏，除此之外，在正常游戏代码的基础上加入自己的创新，比如运用c中文件操作的函数fopen创造出一个类似排行榜的功能，以此来记录不同玩家的分数与记录；在每次操作以后进行清屏，使我们制定的棋盘一直存在于运行界面的上方；添加一个计时功能时刻提醒玩家时间……

2048游戏规则：.

2048是在一个4\*4的正方形格子中进行的，一开始游戏会出现2个数字，在游戏中系统本身只会给出2跟4这2个数字的，而我们的目标是2048。

游戏开始后，玩家要通过移动数字进行合并来达到2048，在游戏中玩家将数字移动一个方向，正宫格中所有的数字也会同样移动，通过这样的移动，2跟2合成4，4跟4合成8以此类推，只有同数值的数字才能合成，我们需要通过一次次的相加最终合成2048。

当玩家成功合成2048后，游戏就顺利通关了，也就是游戏胜利，而如果正宫格中的格子全部填满并且相邻的格子都不相同，也就是无法移动的话，那么游戏结束，游戏失败。

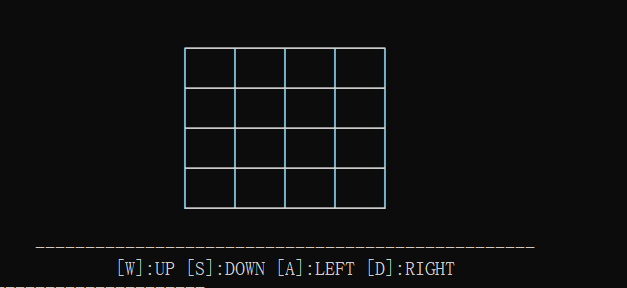
**具体方法说明、示例：**

整个代码的编写可以分为以下 ：

1. 棋盘的制定：

此处我们借鉴了网上的棋盘形式：

<https://blog.csdn.net/qwm8777411/article/details/41322995>

但我们对此进行了修改，使棋盘更加美化样式如下：

并且在显示数字的时候根据数字位数的不同采取不同的打印格式，使其更加美观。

1. 随机数的运用：

此处我们在网上查找函数并进行学习：

<http://c.biancheng.net/view/2043.html>

在这个网址上我们学习到了关于随机数的知识，并用srand函数进行重新播种，使其避免了“伪

随机”，更贴近真正的“随机”。

在代码中我们先寻找空处然后运用随机数确定一个空处生成2或者4（2和4的概率比是1:9），

因此我们对随机数进行求余。

1. 平移代码的实现：

实际上四种移动方式的本质是一样的，只不过是遍历的方式余与顺序不同，而四行的操作可

减为一行的操作。

我们设了两个变量，第一个变量为j，他从1开始遍历直到3；第二个变量为k，他从0开始，他标记的是待合并的下标。可细分为三种情况：

、k和j项相等，对其进行合并，将k处的值赋为合并后的值，然后k++，并将j的值赋为0；

、k项为空，则把j项的值赋给k，但k不进行++操作，并将j的值赋为0；

、k项不为空，但j项不和k项相等，则把j的值赋给k+1的位置，然后k++。

（去掉判断j和k相等的条件）

当这个函数做完后要添加一个新的2或者4。

1. 判断输赢的条件：

赢：

成功的合成了2048。

输：

屏幕全被数字填满，且相邻的格子没有能进行合并的。

1. 对文件进行操作：

运用fopen（text，”r”）和fopen（text，”w”）对文件进行操作，以达到排行榜的作用。

**结果分析：**

**团队分工:**

**源码：**