

西南大学

计算机与信息科学学院

课程设计报告

《C 语言综合课程设计》

题 目： 《2048》游戏

级、专业： 2018 级 计算机类 专业 1 班

学生学号： 222018321212016

学生姓名： 秦凯鑫

指导教师： 刘红俊

提交日期： 2019 年 7 月 17 日

成绩：
评语：

指导教师（签字）：

年 月 日

1 问题描述

运用 C 语言，实现名为《2048》的游戏，从图形画面到后端逻辑用 C 语言来实现，达到可以运行，可以流畅游戏，基本做到无漏洞，无不合理之处。

2 功能需求分析

《2048》是一款简单的游戏，用户在开始后会在一张 4×4 的界面进行游戏，初始时随机两个位置生成 2 或者 4 的两个数字，用户可以通过上下左右，在此为 w、s、a、d 来进行移动，每次按一次键，将进行下列操作：

- 将所有方块向此方向移动一格，已经到达边界的方块不移动；
- 如果目标方向有相同数字的方块，进行合并，将两方块值相加（即为原来 2 倍）；
- 如果目标方向已有方块且数值不同，无法移动；
- 移动完成后在空白处再生成一个 2、4 或 8 的一个方块；
- 如果所有位置充满了方块且此时移动无法消去任意一个方块，则游戏失败
- 若检查已出现数值为 2048 的方块，则游戏胜利。

由上述功能可知，每次合并数值都会乘 2，直到达到 2048，每个数值都为 2 的倍数。

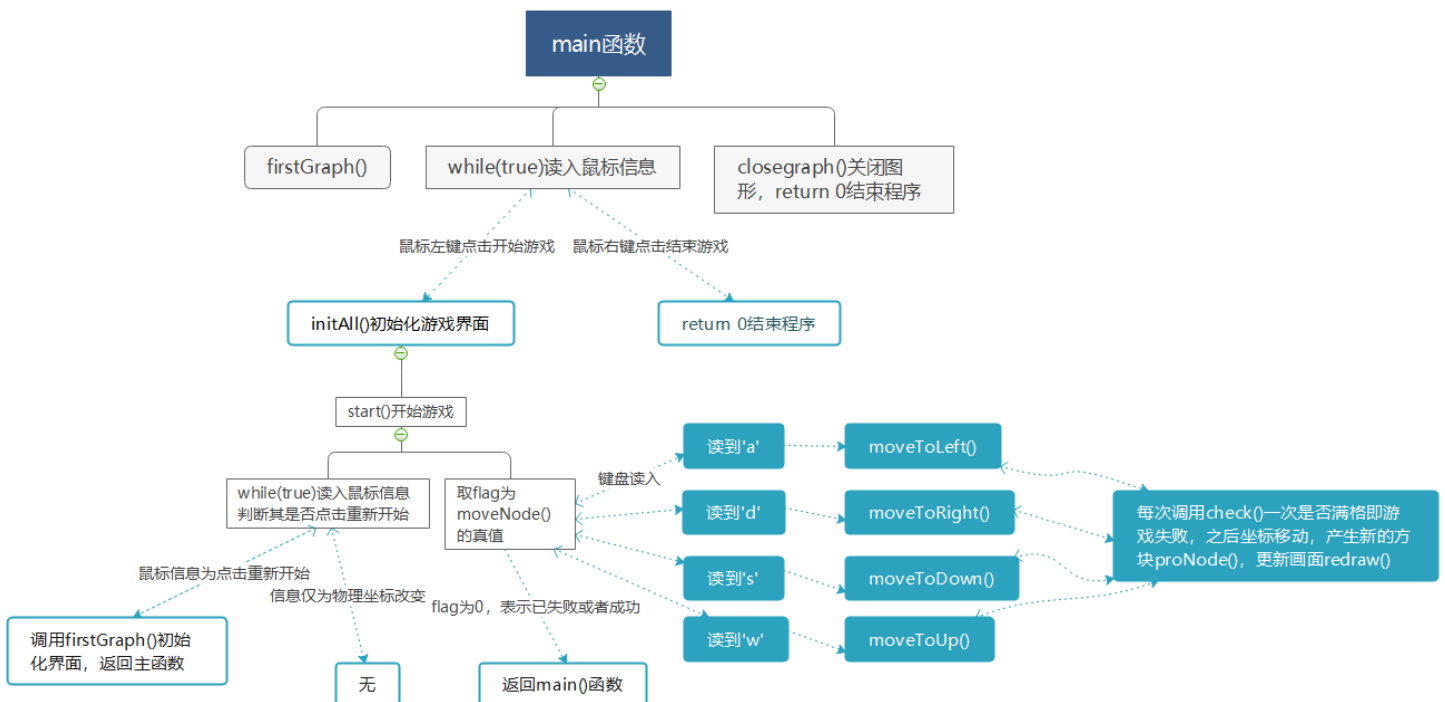
同时还可具备如下功能：

- 首界面可选择开始游戏还是退出游戏，退出即结束。
- 开始游戏后，可选择重新开始，返回主界面，可以重新开始或者退出。
- 右侧有计分项，游戏结束也会有分数提醒。
- 首界面右上角有读档按钮，可以进行读档，同时游戏中可以存档，只需记住存档时所输入 ID 即可读档

3 总体设计

3.1 总体结构

总体结构如下图所示



3.2 函数列表

函数列表如下：

```
void initAll();           //开始后的图形界面初始化
void firstGraph();       //欢迎界面初始化
void reStart();          //重新开始按钮
int moveNode();          //移动
void proNode();          //产生一个随即方块
void redraw();           //更新画面
int check();             //检查格子是否已经满
int moveToUp();          //向上
int moveToDown();        //向下
int moveToLeft();        //向左
int moveToRight();       //向右
int checkWin();          //检查是否达到2048
void start();            //开始
void TcharToChar(const TCHAR* tchar, char* _char); //TCHAR转char
int Login();             //登陆选项
void deleteN(char*);     //删除换行符
void dataFind();         //文件中数据找回
int preserve();          //储存档案
```

3.3 整体思路

定义结构体 Node，成员变量为 x，y，num，分别表示一个块的 x 坐标，y 坐标以及当前所存数字，每次移动时扫描除了此 Node 外的其他 Node，出现目的坐标的点时判断其 num 是否相同，若相同则清空被移动 Node 信息，将目的坐标 Node.num 乘以 2，通过再次绘图可以得到点的移动。

分数 getScore 是每次合并即可加一分，同时定义结构体 Color，以 rgb 形式存储颜色，定义 bColor 和 cColor 分别表示背景颜色和字体颜色，由于方块背景只有 16 种即可定义 16 个 bColor，字体颜色由于 2-2048 共 11 个数字对应了 2 的幂次，则可只定义 11 种以 log 求值来索引颜色。

data.txt 中，以一行存储用户输入 ID，之后紧跟第二行两个整数，分别表示 nodeNum 和 getScore，再往下即 nodes 中所存信息 x，y，num，每三个一行，直到下一个 ID，文件读写时用 fopen 打开，fgets 读和 fputs 写完成。

首先进入 main 函数，通过 initgraph() 创建绘图环境，再进入 firstGraph()，通过一系列绘图函数，加载图片等，创建出首界面，首界面包含了项目名称，“开始”和“结束”以及“读档”按钮，开始和结束按钮下附带有简单的游戏说明，点击开始按钮后会进入到游戏界面，main 函数通过 while(true) 来控制鼠标传入信息是物理坐标的改变还是按压左键，当左键单击按钮时，调用 initAll() 函数来初始化游戏界面，初始化完成后，进入 start() 函数即可正常游戏，中途若想停止或者重新开始，可鼠标点击右侧存档按钮和重新开始按钮，存档时调用 preserve() 函数，点击重新开始按钮将会重新绘制首界面，然后返回主函数。

在首界面，若点击读档按钮，将调用 Login() 函数，函数将会提示用户输入 ID，获取到 ID 后，调用 dataFind()，尝试从 data.txt 中查找数据开始游戏。

首界面如下：



4 详细设计

4.1 main() 函数

在 main() 函数中初始画布，调用 firstGraph() 来绘制首界面，while (true) 可以实现持续读入鼠标信息，只有当鼠标信息为左键按下时，才会有其他行为，读入鼠标物理坐标位移时，并不会触发其他函数。当鼠标左键单击“开始”区域对应的坐标范围时，调用 initAll() 初始化游戏界面，再调用 start() 开始游戏，当鼠标左键单击“结束”区域对应的坐标范围时，直接 return 0，结束程序，当鼠标左键单击“读档”区域对应的坐标范围时，程序调用 Login() 并进行判断，Login() 返回 0 说明用户选择取消读档，返回 1 则说明读档成功，调用 dataFind() 函数读取 data.txt 中的信息，此时 nodes 中所存节点信息即存档信息，只需调用 redraw() 即可复原之前画面，再调用 start() 即可正常游戏，循环结束后 closegraph() 关闭画布。

4.2 firstGraph() 函数

通过 EasyX 的图形库，loadimage() 加载背景图，outtextxy() 在指定范围内显示字体来完成首界面的初始化，其中因为游戏失败或成功时也会调用此函数，因此加上 memset 将 nodes 清空。

4.3 initAll() 函数

初始化游戏界面，游戏界面由两部分构成，一部分为左边 4×4 的方块阵，右边为菜单即分数显示，菜单可选择存档和重新开始。

左边方块阵利用 EasyX 的 fillrectangle 画出图形，两重 for 循环加上坐标计算即可实现，再通过以时间为种子的随机数模拟，生成两个随机位置（不重复），随机是 2 或 4 的两个方块，由于以像素为单位的坐标不容易对比，因此 Node 中的 x, y 都存的是第几个方块，通过数学计算算出合适的像素坐标。

4.4 start() 函数

start() 函数同 main() 函数一样，以 while(true) 控制鼠标或键盘信息读入，如果鼠标点击了菜单中的重新开始按钮，则调用 firstGraph() 重新开始并 return 到 main() 函数里，

重新选择，若鼠标点击了存档，则调用 `preserve()` 函数并判断，返回值为 0 表示用户选择取消输入，则 `continue`，返回值为 1 表示用户存档成功，调用 `firstGraph()` 函数，并 `return` 到 `main()` 函数重新选择。

顺序结构调用 `moveNode()` 函数，判断是否读到键盘信息，若未读到键盘信息，则返回 `main()` 函数，若读到，读入信息后进行判断上下左右，判断前要调用 `checkWin()` 函数检查是否已有 `num` 值未 2048 的方块，若没有，进行下面的上下左右（分别为键 `w`、`s`、`a`、`d`）判断，对应不同函数调用，并判断 `moveToXXX()` 四个函数的返回值，若为 0，则说明此时游戏出现失败，则需要调用 `restart()` 重新开始，若有 2048，则会通过 `checkWin()` 弹出获胜字样并调用 `restart()` 绘制重新开始按钮。

4.5 `moveToLeft()` 函数

以 `moveToLeft()` 函数为例来说明，其余同理。函数会先调用 `check()` 函数判断一下，是否已经有 16 个相邻不同值的方块因此不能移动，若 `check()` 值为 0 说明不能移动，`moveToLeft()` 函数返回值 0，移动失败，若返回值为 1，则进行移动，利用两重循环，通过对 `Node` 的遍历，可以知道当前块的目标位置是否有方块以及值是否相同，设立变量 `flag`，为 0 则记为目标位置为空，1 表明目标位置有方块，最后判断 `flag` 的值，若为 0 则对当前移动方块的 `x--`，函数最后利用 `proNode()` 函数在空白地产生随机 2、4 或 8 的随机方块，并利用更新的 `nodes`，调用 `redraw()` 重新绘制整个图，然后返回 1 表明移动成功。

4.6 `Login()`，`dataFind()` 以及 `preserve()` 函数

当在首界面点击读档时，调用 `Login()` 函数，函数会通过 `graphics.h` 库的 `InputBox` 弹出窗口，提醒用户输入 ID 以读档，若用户取消输入，`InputBox` 返回值为 0，此时 `Login` 返回值也为 0 且直接返回，若为 1，将用户输入的 `Identity` 调用函数 `TcharToChar()` 转换成 `char` 字符串类型，返回 1。

若 `Login()` 函数返回值为 1，将会调用 `dataFind()` 函数在 `data.txt` 里查询信息读档，`dataFind()` 函数通过对字符串及文件读写的操作，`fopen` 打开文件，通过 `fgets` 读一行，对用户的 `Identity` 进行对比，利用 `strcmp()` 函数可快速对比，若 ID 一致则进行数据复原，将储存信息赋值到 `nodes` 中，然后可以不用继续往下找，`break` 即可。

若用户在游戏中点击存档，即在 `start()` 中调用了函数 `preserve()` 来存档，`preserve()` 函数以 `a+` 模式打开文件，要求用户输入一个 ID 并以 `while` 来重复判断，每次输入完后会查重，若 `data.txt` 中已存在此 ID，则确定后会重新弹出，直到不重复为止，若用户取消输入，则返回真值 0。与 `dataFind()` 相反，`preserve()` 会在文件打开后将已经转换成字符串的 `x`，`y`，`num` 值 `fputs` 进文件中，最后 `fclose()` 关闭，返回真值 1。

5 系统开发

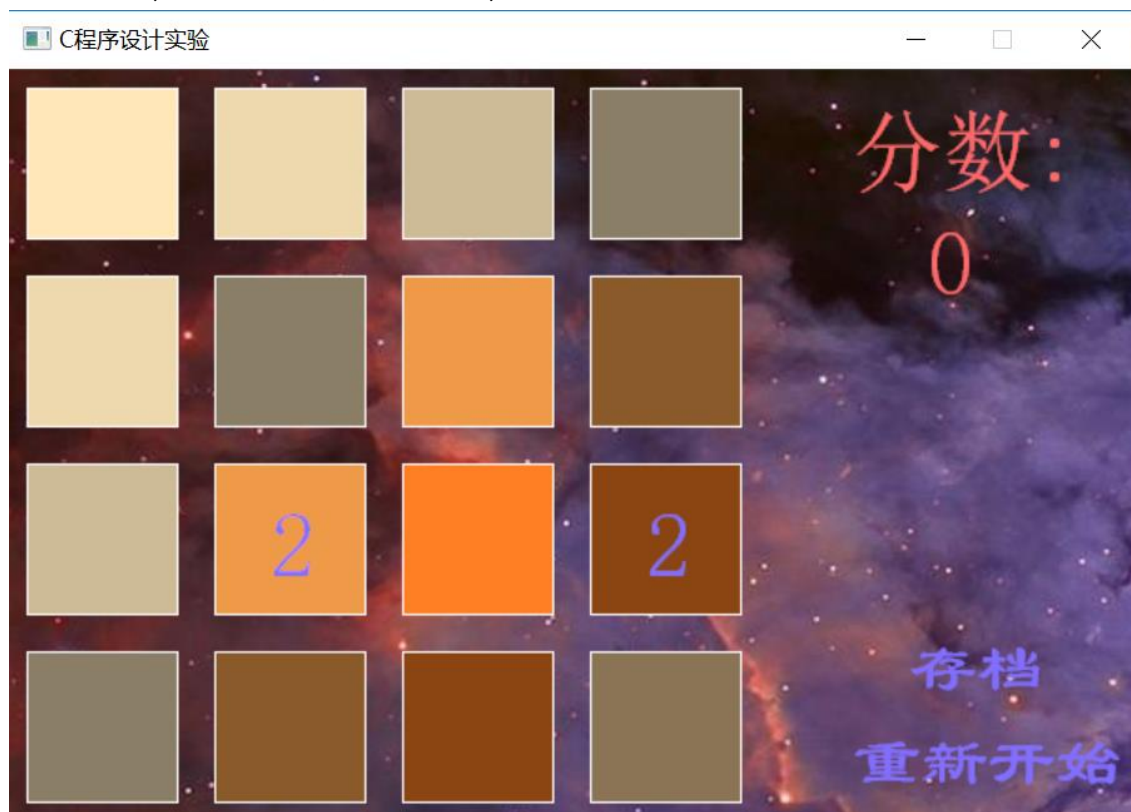
程序采用 VS2019 实现，编译及运行，运用第三方库 `EasyX` 绘图。

6 测试

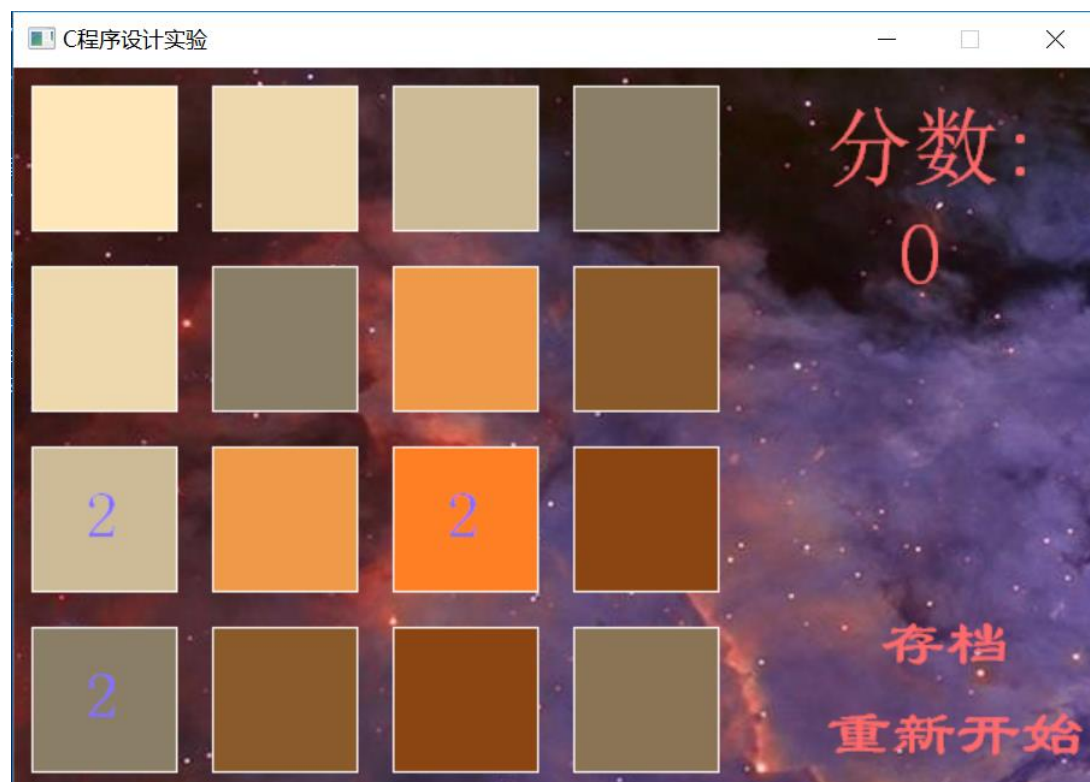
通过测试，基本功能都可实现，程序执行截图如下：
首先是首界面，



点击开始后，将进入游戏界面的初始化，



左移（a）后，



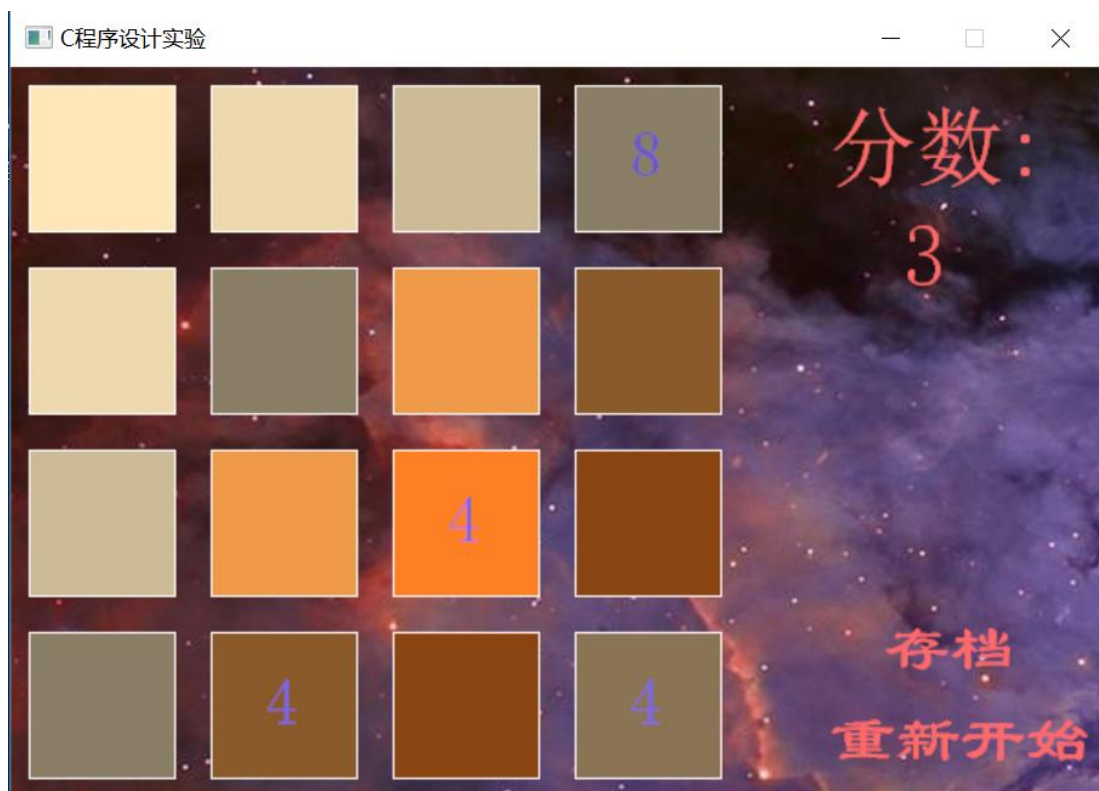
上移 (w) 后,



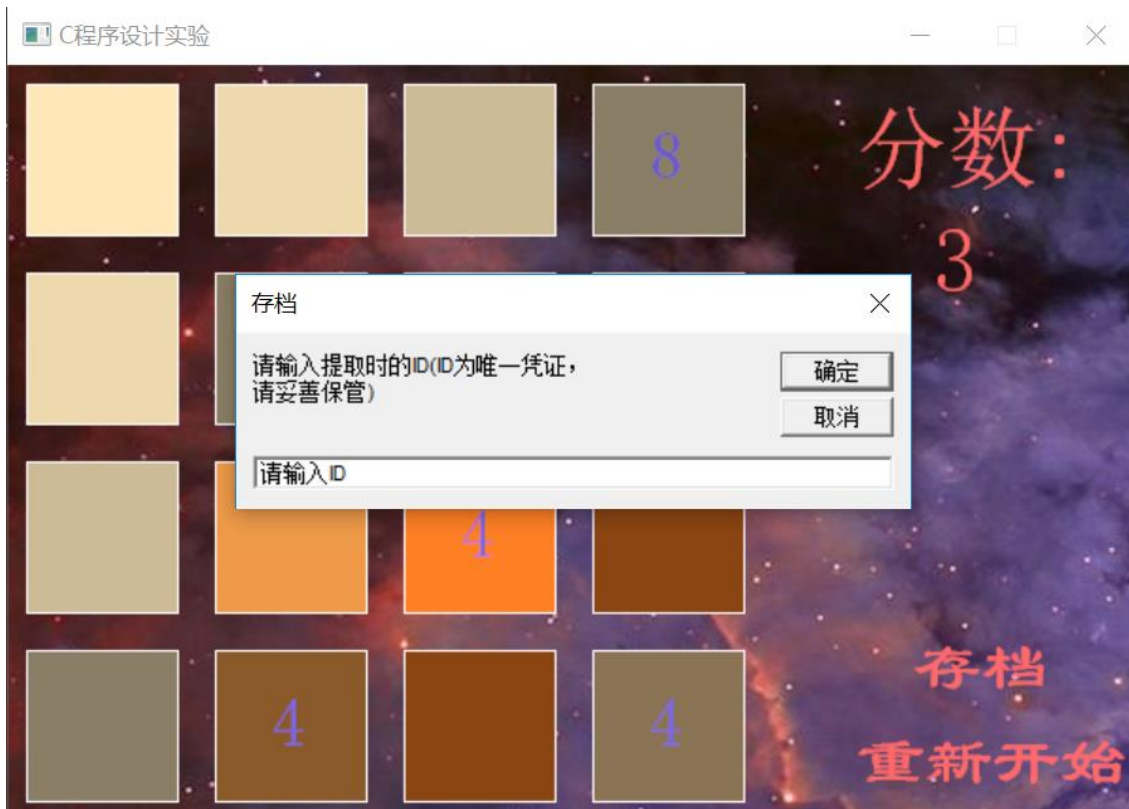
下移 (s) 后,



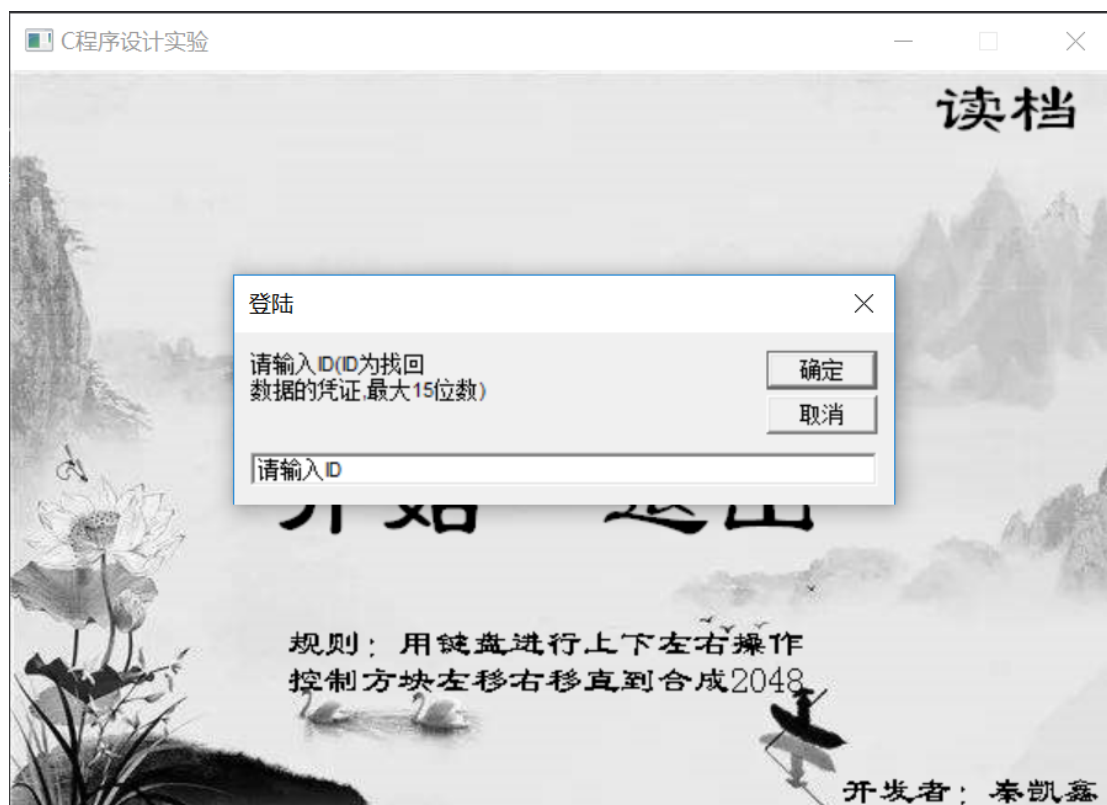
右移 (d) 后,



点击存档后，



点击读档后，



游戏失败，



游戏成功，



7 参考文献

- [1] Stephen Prata. 《C Primer Plus》 [M]. 北京:人民邮电出版社,2016.
[2] 耿祥义, 张跃平. 《C 程序设计教学做一体化教程》 [M]. 北京:清华大学出版社, 2013.

附录:

已将作业整体上传于 github, 欢迎老师查阅

<https://github.com/qkxwxy123/2048>

课程设计报告评级指标如下:

评价指标	综合考虑左侧评价指标, 参考毕业论文的成绩评定方法, 给出 5 分制的课程论文成绩。
报告符合规定的格式	
模块的划分合理	
数据结构和算法的时空性能高	
图表规范	
代码符合规定的风格要求	