

浙江大学

程 序 设 计 专 题

大 程 序 报 告



大程名称： 俄罗斯方块

课程名称： 程序设计专题

姓 名 1: _____
姓 名 2: _____
姓 名 3: _____
学 号 1: _____
学 号 2: _____
学 号 3: _____

指导老师： 王建江

2020~2021 春夏学期 2021 年 7 月 7 日

目 录

1	大程序简介	3
1.1	选题背景及意义	3
1.2	目标要求	4
2	需求分析	5
2.1	功能需求	5
2.2	数据需求	6
3	程序开发设计	7
3.1	总体架构设计	7
3.2	功能模块设计	7
3.3	数据结构设计	8
3.4	源代码文件组织设计	9
3.5	算法设计描述	10
4	部署运行和使用说明	12
4.1	编译安装	12
4.2	使用操作	12
5	团队合作	13
5.1	任务分工	13
5.2	开发计划	14
5.3	收获感言	14
6	合作总结	17
6.1	合作记录	17
6.2	知识点总结	17
7	诚信说明	17

俄罗斯方块大程序设计项目

1 大程序简介

1.1 选题背景及意义

为进一步熟悉 c 语言和图形库函数，提升组员们的编程能力，将所学知识有效地应用到实际当中，经过讨论，我们三人决定选择俄罗斯方块大作业。通过这次作业可以培养我们的逻辑能力和自学能力。在完成作业的过程中学会小组合作完成一个工程项目，合理地分工、构建框架，帮助我们更好地理解回调函数、全局变量、链表等知识，实现从控制台到图形界面的转变。

俄罗斯方块(Tetris, 俄文: Тетрис), 基于 libgraphics, 设计和实现俄罗斯方块游戏。它是一款风靡全球的电视游戏机和掌上游戏机游戏, 由俄罗斯人阿列克谢·帕基特诺夫发明, 故得此名。俄罗斯方块的基本规则是移动、旋转和摆放游戏自动输出的各种方块, 使之排列成完整的一行或多行并且消除得分。由于上手简单、老少皆宜, 从而家喻户晓, 风靡世界。

1.2 目标要求

- (1) 实现用户系统，能够打开和保存游戏。
- (2) 游戏能顺利运行，实现游戏功能。
- (3) 游戏可选难度等级，实现基本操作，如暂停、重来、退出等。
- (4) 游戏能够在窗口实时显示操作结果、下一方块和当前分数。
- (5) 需设置快捷键响应。
- (6) 能从文本文件中读入用户信息和最高分，格式为：<用户名> <最高分>

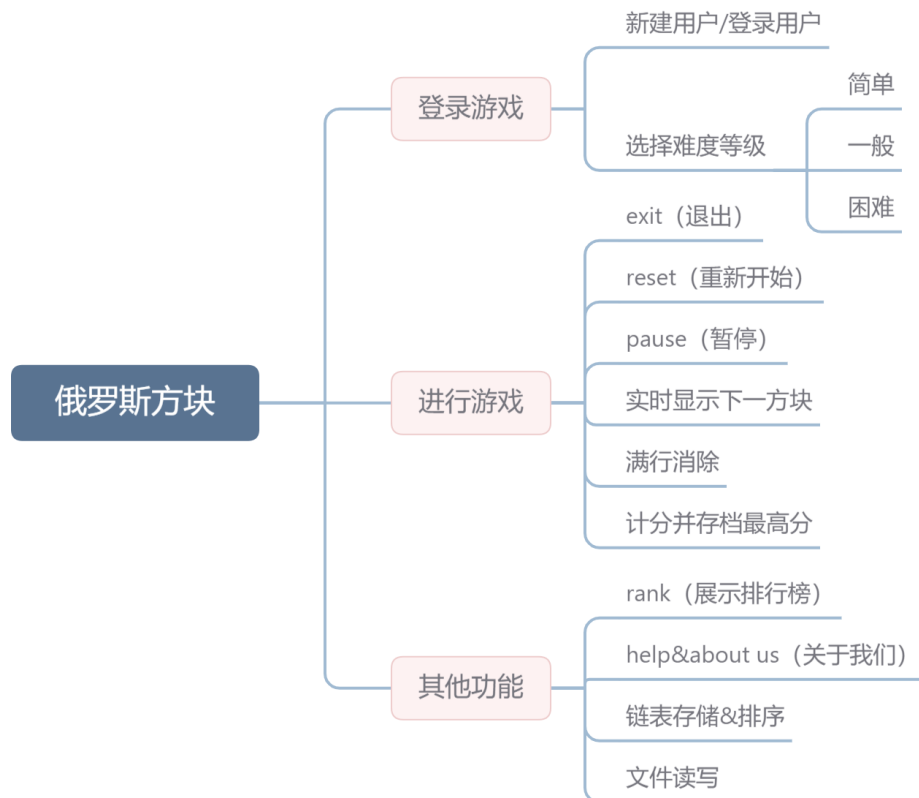
例如：



- (7) 支持过程的存档和调取，即能够在演示过程中，1) 存储当前游戏用户信息（包括用户名及最高分）2) 用图形界面展示排行榜 3) 利用 `system` 命令打开文本文件。

2 需求分析

2.1 功能需求



(1) 本程序为俄罗斯方块，目的在于实现游戏功能。本程序分三个界面：初始登录界面、主界面和排行榜界面。主界面主要有 4 个按钮，分别为“exit”“reset”“pause”“about us”。程序运行时先在登陆界面输入用户名，选择用户难度再进入游戏界面，游戏中可随时查看排行榜并返回。

(2) 本程序初始界面登录用户后有三种难度选择，对应三种不同的方块下落速度。进入主界面后游戏自动开始，使用键盘上方向键旋转方块、下方向键加速下落、左右方向键控制方块左右移动。

(3) 游戏中可以点击右侧“exit”按钮退出程序，“reset”按钮重启游戏，“pause”按钮暂停方块运动，“about us”按钮查看帮助。点击左上方菜单栏“rank”选项，在下拉菜单中选择“open”（或使用 Ctrl-0）打开排行榜，“close”（或使用 Ctrl-C）关

闭排行榜返回游戏界面（仅在排行榜界面中生效）；点击“tool”选项，在下拉菜单中选择“reset”（或使用 Ctrl-R）“pause”（或使用 Ctrl-P）“exit”（或使用 Ctrl-E）实现对应功能；点击“help”选项，在下拉菜单中选择“About Us”（或使用 Ctrl-A）打开游戏帮助。

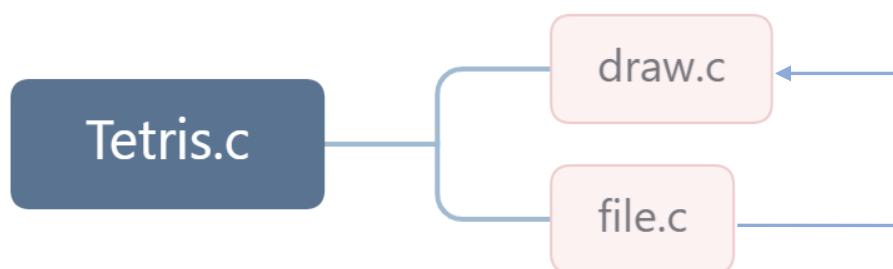
2.2 数据需求

（1）俄罗斯方块会自动从“俄罗斯方块.txt”文件中读取之前存档的游戏用户及分数。同时，游戏中也会自动保存当前用户及游戏最高记录，并进行排序显示在排行榜界面。

（2）程序读取时采用动态分布，利用链表结构来读取和存储数据。

3 程序开发设计

3.1 总体架构设计



3.2 功能模块设计

- 1.功能主要分为用户登录、难度选择、游戏主体、分数存储、排行榜显示。
- 2.数据从 txt 文本文件以文件的方式读取，并存储至链表中，用户登录后存储当前用户信息，中途关闭程序或退出游戏时，将此用户得分存档至 txt 文件中，下一次打开时，可以读取之前所有的数据。
- 3.登录界面可以自主选择游戏难度。
- 4.左边游戏界面实时显示游戏状态，右边实时显示下一个方块和当前分数。

3.3 数据结构设计

- (1) 整型变量及整型数组，双精度浮点型变量：

```
double winwidth, winheight; //窗口长度、宽度
double eex, eey; //当前方块原点坐标
int board[11][21]; //存放坐标是否填充
int color[11][21]; //存放坐标的颜色
int score; //记录分数
int enter = 0; //判断是否进入游戏
int rank = 0; //判断是否打开排行榜界面
int mseconds; //定时器调用时间
int t = 0, r = 0; //按钮
int i; //当前的方块序号
int next; //下一个方块序号
int enable_mov_Tetris = 1; //允许方块移动
int if_fall = 0; //标记能否下落
int anti_keyboard = 1; //在判断满行时不执行键盘操作
int c = 0; //判断是否选择了难度等级
int flag = 0; //判断是否按了exit（如果按了就保存成绩后退出）
char name[80]; //存储当前用户名
struct link *p = NULL; //链表头节点
```

- (2) 结构体及链表：

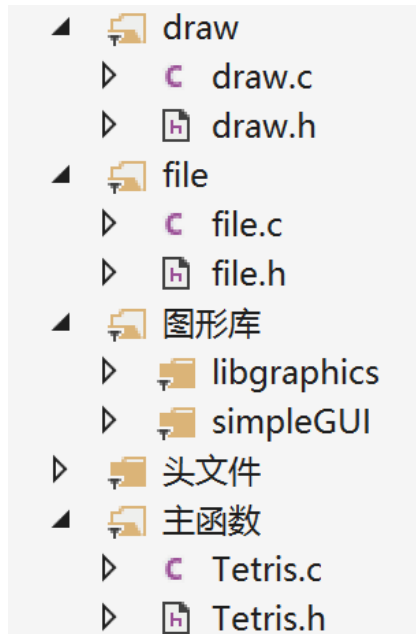
```
struct BLOCK //俄罗斯方块数据
{
    double x1; //坐标
    double y1;
    double x2;
    double y2;
    double x3;
    double y3;
    double x4;
    double y4;
    char color[8]; //颜色
    int num; //方块编号
};

struct link { //链表结构
    char name[80];
    int score;
    struct link *next;
};
```


3.4 源代码文件组织设计

<文件目录结构>

1) 文件函数结构



draw.c

//初始界面

void drawTopic();//开始界面—绘制标题，登录

void ChooseLevel();//选择难度等级

void displaychar();//画用户名的输入框

void displaystart();//画登录界面的确认按钮

//游戏界面

void DrawMoving(double x, double y);//画当前的方块

void DrawSilent();//绘制已经落下的方块

void DrawLattice();//画网格

void DrawBackground();//画背景

void DrawNext(int i);//在右边画生成的下一个方块

void DrawScore();//显示分数

void DrawButton();//画按钮

void DrawMenu();//画上方菜单栏 快捷键

void display();//显示

//排行榜界面

void DrawRank();//显示排行榜

file.c

```
struct link *SaveData(struct link *p); //保存最高纪录并写入文件
struct link *initLink(); //从文件读取数据建立链表
struct link *insert(struct link *p); //插入新用户（采用头插）
struct link *datasort(struct link *head); //冒泡排序（交换节点法）
```

Tetris.c

```
//回调函数
void TimerEventProcess(int timerID); //定时器
void myKeyboardEventProcess(int key, int event); //键盘回调函数
void myMouseEventProcess(int x, int y, int button, int event); //鼠标回调函数
void CharEventProcess(char ch); //获取输入的字符

//游戏函数
int Transferx(double tipx); //把界面的x坐标（double）转换成数组的x坐标
int Transfery(double tipy); //把界面的y坐标（double）转换成数组的y坐标
int FullLine(); //判断满行
void KillLine(int); //清除底行
int TouchLeftWall(double x, int num); //判断如果左移会不会触到左边界
int TouchRightWall(double x, int num); //判断如果右移会不会触到右边界
int GameOver(); //游戏结束

//初始化
void initRandomNum(); //随机初始化
```

3.5 算法设计描述

（1）游戏主体

①每次用户注册或登陆后，游戏开始，同时打开定时器；

使用一个数组存储用户名，用户通过键盘输入用户名时存储在数组中，刷新界面，将用户名实时显示在屏幕上。通过改变定时器的定时时间来实现不同的难度等级，定时时间越短游戏时方块下落得就越快。当用户按下确认按钮时调用游戏主界面绘制函数进入游戏，并打开定时器。

②每一小轮游戏的迭代：由 display（）函数控制游戏界面的刷新显示，每次刷新重新绘制游戏界面，包括网格、菜单栏、功能键和下一个方块 Next、得

分 Score 的绘制，以及正在移动的和已落下的方块的绘制，当方块可移动（不越过网格边界并且不触碰已落下的方块）时，可由相应功能键实现方块的左移、右移、旋转和加速下落，当判断方块不可再下落（触碰下边界或下一行已有方块）时，由 `board[][]` 和 `color[][]` 数组存储该方块的信息，并判断是否有方块满行，如有则实现消行，分数加 **100**，然后随机生成一个新的方块，进行新一轮的移动控制和判断。

③每一大局游戏的迭代：当已落下的方块堆积的高度超过设定值时，为一次游戏的结束，存储对应的成绩 Score，用户可选择开始新一局的游戏。

（2）排行榜（链表）

判断当前用户是否为新用户，若不是则将用户作为新结点插入链表（采用头插的方式）。完成数据组织后对链表数据以 score 为关键字进行冒泡排序，再画出排行榜界面，调用绘制菜单函数保留菜单栏。

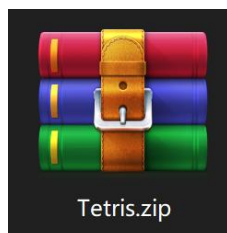
（3）文件读写

通过 `fscanf` 的格式化输入读取存档数据，以链表形式动态存储。在游戏的几个节点（如退出、游戏失败、打开排行榜切换界面等）自动保存当前用户的信息，调用链表主函数存储并排序。

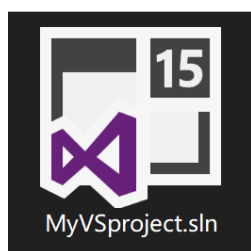
4 部署运行和使用说明

4.1 编译安装

1. 下载 VS2017
2. 下载压缩包 Tetris.zip



3. 解压压缩包，打开\Tetris\MyVSproject\MyVSproject.sln



4.2 使用操作

1. 在登录界面需要先输入用户名再选择难度。输入用户名时，需要敲击一次回车无法读入最后一个字符，需要再敲击一次。
2. 进入主界面后游戏自动开始，网格中的红线是游戏失败的判定线。使用键盘上方向键旋转方块、下方向键加速下落、左右方向键控制方块左右移动。游戏过程中可随时点击右侧按钮和左上方状态栏进行操作。
3. 游戏失败后会有弹窗选择重来或者打开排行榜，从排行榜返回后游戏重新开始。

5 团队合作

5.1 任务分工

:

- 1、游戏主体部分框架搭建
- 2、游戏主体部分函数编写、修改
- 3、绘制整个登录界面，包括回调函数、读取用户名存储、选择难度等级等
- 4、绘制按钮实现响应
- 5、写入单个用户文件

:

- 1、游戏主体部分框架搭建
- 2、游戏主体部分函数编写、修改
- 3、键盘回调函数的响应
- 4、绘制菜单栏
- 5、快捷键响应

:

- 1、游戏主体部分框架完善
- 2、游戏主体部分函数编写、修改
- 3、整理代码及注释，完成多文件架构
- 4、链表结构存储
- 5、实现链表与文件读写
- 6、绘制排行榜界面
- 7、整体修改UI及配色
- 8、制作PPT

5.2 开发计划

- 1.先各自学习了图形库文件。
- 2.小组进行了一次讨论，进行了思路框架的整理，将各个模块列出，并做了简要的分工，对各自学习研究的方向有了更深入的了解。
- 3.共同合作，共同撰写主题代码，相互分享、修改出现的 bug，实现了游戏的主要功能。
- 4.**同学完成登陆界面的绘制（至此已实现游戏基本功能）。
- 5.（因为考试周结束得早）**同学完成链表结构和文件读写，修改程序框架完成多文件架构，实现排行榜显示。
- 6.整理代码及注释，使函数及结构更清晰。
- 7.制作 PPT，撰写报告。

5.3 收获感言

总体描述

本次程序设计大程序开发中我们前期花了大量的时间去学习图形库，但程序上还是有一些比较细小的问题以及不足。我们在构建整个框架的时候考虑得也不够周到，在已有基础上不断增加新的功能。在写程序的时候不够规范，有时候没有及时地加上注释，组员在读对方编写的程序时就会十分地费劲。不过在后期，组员们提高了作业效率。我们参考了 b 站、CSDN 等网站上对俄罗斯方块功能的实现。

在这次项目中，我主要负责总体框架的搭建和登录界面的实现。在开始初期，由于对图形库不了解完全不知道如何下手，也难以静下心来认真学习，导致开始的时间比较晚，后面也比较赶。不过通过组员的帮助以及对例程的学习，我渐渐开始理解图形库中的函数，也慢慢学会了参照例程使用函数实现自己想要的功

能。在登录界面的实现中，绘制输入框、难度等级的选择、进入游戏的判断、用户的存储等都花了比较长的时间。在经历一次又一次的失败后，终于在运行后看到了想要的样子，内心还是很兴奋的。

通过这次大作业，我明白了万事开头难，开始做一件自己不熟悉不了解的事情是很困难的，须组员这是一个必经的阶段，只有静下心来认真钻研学习，不怕麻烦，才能为后面打好基础，提高效率。也更加理解了之间团结的重要性，及时的沟通合作可以有效减少麻烦，在。这次大作业对我来说是一次不小的挑战，学习 C 语言的时间隔了比较久，很多东西都要重新去回顾，还要学习之前没有接触过的图形库，经过这次挑战，我对 C 语言的理解更深了，看到自己写的一行行代码运行出了自己期望的结果，还是比较有成就感的。

在这次程序的编写过程中，我主要负责的是总体框架的搭建和功能键的实现。虽然对 C 语言基础有了一定的学习，但我发现自己对 C 语言的理解和应用还是有很多不足的。

这次大作业暴露出我的一些问题，但也让我的能力得到了提升。一方面我们小组在前期的规划不够清晰，因此在后期交流工作时出现了一些问题，不过感谢组员的包容和帮助，我们最终还是互帮互助，一起完成了这项任务，这次作业也给了我一个重新反思审视自己的机会；另一方面，通过这次大作业，我也更加坚定了没有克服不了的困难，只要静下心来，认真钻研，总会有“柳岸花明”的一刻。同时，团队的合作将产生远大于个人的力量，虽然前期由于对图形库不熟悉，完全不知道如何入手，但在沉下心来认真学习后，我们小组成员合作将程序的大体框架搭建起来，这之后好像思路一下子就开阔了，每个成员接力完善程序的各个功能，接下来的工作也顺利得多了。

在这次大作业中，我主要负责文件的读取、存档和调用，链表结构以及排行

榜功能的实现。在游戏主体的共同合作中主要实现了最初的定时器功能。因为考试周结束得比较早，在修改整理最终代码后制作了宣讲 ppt。我认为在这一次作业中我收获了许多，也学到了许多。

首先，我认为这次作业对自己的耐心与细心都有很大的提升，在前期规划中，我们小组的安排不够清晰，没有厘清项目的总体框架，以至于在大家的协作中暴露了一些问题与不足。后来我们找到了相互合作的节奏，效率也提高了很多。因为考试周的缘故，我负责了后期大部分内容。在此次作业中我了解并学会了图形库基本操作，在研究文件读写的过程中，不断学习调试，我设置断点、查看变量，控制变量排除每个函数的问题，最后成功解决，对文件这块知识也有了更深层次的理解，掌握得也更加全面。在进行多文件架构的过程中，修改中出现了无数各种各样的 bug，我在修改的过程中自学查询各种错误的原理和解决办法，最后完成了多文件架构。事后总结发现多文件架构本应在分工时完成，分别完成不同部分后再整体组合。

在写大作业的过程中我体会到了代码的结构性，与平时做题需要的算法不同。我还体会到小组合作的重要性，小组成员之间的互相帮助使得许多错误与问题得以更快地解决，互相讨论中发现我无法解决的问题可能在别人手中迎刃而解，看到别人的代码也会有恍然大悟的感觉。

6 合作总结

6.1 合作记录

6.2 知识点总结

回调函数、图形库应用
实际坐标与数组存储的转换
多文件结构及全局变量
链表存储、插入、排序（交换节点）

7 诚信说明

依照学术诚信条款，我们保证本次大作业没有全盘抄袭他人作品。