

《程序设计训练》大作业

俄罗斯方块



组员姓名： 郭龙威 刘艺 卢仪

吴雅婷 孙逸飞

院系所属： 电子信息和电气工程

专业班级： 计日 1 4 0 2

指导教师： 齐 恒

《程序设计训练》大作业

俄罗斯方块

目录

一.程序设计内容.....	2
A.问题概述.....	2
B.基本功能.....	2
C.设计目的.....	2
二.小组成员构成.....	2
三.程序功能与分块.....	3
A.界面设计.....	3
B.设计思路.....	3
C.头文件.....	3
D.全局变量.....	3
E.重要函数及功能描述.....	4
1.方块组合.....	4
2.改变颜色 void ShowSquare2(HDC hMemDc).....	5
3.成线消除 void DestroyOneLineSquare().....	5
4.旋转图形.....	6
5.其他函数.....	6
四.心得体会.....	7

用 C 语言编写简易版的俄罗斯方块游戏，从而巩固对 C 语言知识的掌握和实现对其灵活运用，学习程序设计的基本思想与方法。

关键词：C 语言，俄罗斯方块，基本图形绘制，矩阵，平移，旋转

一.程序设计内容

A.问题概述

用 C 编写俄罗斯方块游戏，使游戏能够在 Microsoft Visual C++6.0 上运行。

要求支持键盘操作，有 7 种不同类型方块的旋转变换，界面显示当前玩家的得分及等级。随着游戏的进行，等级越高，相对应游戏难度应越大（即方块的下落速度变快），消去一行所得到的分数越高。

B.基本功能

任意键开始游戏；

随机产生方块并自动下移；

方向键控制游戏：↑键控制方块变形、↓键控制方块下移、
←键控制方块左移、→键控制方块右移；

能正确判断满行并消行；

能正确计分并判定级别：每消除一行获得 **100** 分，达到一定分数自动可提高一等级；

游戏结束界面：包括继续游戏选项和退出选项。

C.设计目的

本项目以督促进学生复习巩固由 C 语言和数据结构作代表的计算机程序设计知识为中心，加深对程序设计方法、技术以及思想的理解与感悟，充分追求将所学知识运用到实际开发中，发扬从实践中来到实践中去的思想。

藉此，希望能够使得学生初步掌握程序设计的基本方法和流程，同时寓教于乐，鼓励积极开展团队合作，以全方位地提高学生的综合应用能力。

二.小组成员构成

基本信息：所属院系：电子信息与电气工程

专业班级：计日 1402

组员构成：郭龙威（201487017）

刘 艺（201487019）

卢 仪（201487034）

吴雅婷（201487035）

孙逸飞（201487037）

（本项目由粗体标出的人员负责答辩）

三.程序功能与分块

A.界面设计

右侧状态栏 157-154-163

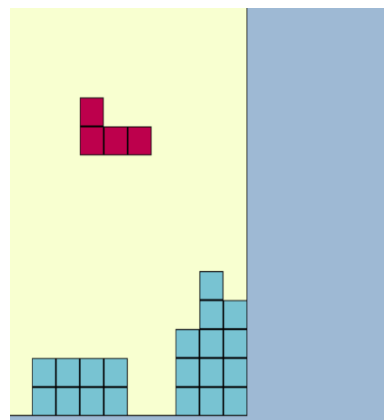
左侧背景 246-255-204

方块(下落中) 184-0-71

方块(落地时) 115-190-206

等级和得分 0-0-0

等级背景 255-255-255



游戏主界面：宽 10、高 20 的游戏板

(在此游戏中，将 30 个像素定义为一个方格的边长)

B.设计思路

显示玩家初始的成绩，初始成绩为 0，初始速度 550(ms)；

每消除一行可得 100 分，每获得 500 分可进入下一级，即下落速度提升 50 毫秒，等到下降速度为 50 毫秒时不再加快下降速度，但仍可累计得分晋级；

有简单可行的操作（方向键控制）；

有合理判断游戏结束条件，并能进行退出。

C.头文件

来自系统 `#include <windows.h>`

`#include <time.h>`

自我演绎 `#include "elshead.h"`

D.全局变量

```
char g_arrBackGround[20][10] = {0};
```

```
char g_arrSquare[2][4] = {0};
```

```
int g_nSquareID = -1;
```

```
int g_nLine = -1;
```

```
int g_nList = -1;
```

```
int g_nScore = 0;
```

```
int g_nLevel = 1;
```

```
int timeCounter = 550;
```

E.重要函数及功能描述

1.方块组合

代码	形状
<pre>g_arrSquare[0][0]=1, g_arrSquare[0][1]=1, g_arrSquare[0][2]=0, g_arrSquare[0][3]=0; g_arrSquare[1][0]=0, g_arrSquare[1][1]=1, g_arrSquare[1][2]=1, g_arrSquare[1][3]=0; g_nLine = 0; g_nList = 3;</pre>	
<pre>g_arrSquare[0][0]=0, g_arrSquare[0][1]=1, g_arrSquare[0][2]=1, g_arrSquare[0][3]=0; g_arrSquare[1][0]=1, g_arrSquare[1][1]=1, g_arrSquare[1][2]=0, g_arrSquare[1][3]=0; g_nLine = 0; g_nList = 3;</pre>	
<pre>g_arrSquare[0][0]=0, g_arrSquare[0][1]=1, g_arrSquare[0][2]=0, g_arrSquare[0][3]=0; g_arrSquare[1][0]=1, g_arrSquare[1][1]=1, g_arrSquare[1][2]=1, g_arrSquare[1][3]=0; g_nLine = 0; g_nList = 3;</pre>	
<pre>g_arrSquare[0][0]=1, g_arrSquare[0][1]=0, g_arrSquare[0][2]=0, g_arrSquare[0][3]=0; g_arrSquare[1][0]=1, g_arrSquare[1][1]=1, g_arrSquare[1][2]=1, g_arrSquare[1][3]=0; g_nLine = 0; g_nList = 3;</pre>	
<pre>g_arrSquare[0][0]=0, g_arrSquare[0][1]=0, g_arrSquare[0][2]=1, g_arrSquare[0][3]=0; g_arrSquare[1][0]=1, g_arrSquare[1][1]=1, g_arrSquare[1][2]=1, g_arrSquare[1][3]=0; g_nLine = 0; g_nList = 3;</pre>	
<pre>g_arrSquare[0][0]=1, g_arrSquare[0][1]=1, g_arrSquare[0][2]=0, g_arrSquare[0][3]=0; g_arrSquare[1][0]=1, g_arrSquare[1][1]=1, g_arrSquare[1][2]=0, g_arrSquare[1][3]=0; g_nLine = 0; g_nList = 3;</pre>	
<pre>g_arrSquare[0][0]=1, g_arrSquare[0][1]=1, g_arrSquare[0][2]=1, g_arrSquare[0][3]=1; g_arrSquare[1][0]=0, g_arrSquare[1][1]=0, g_arrSquare[1][2]=0, g_arrSquare[1][3]=0; g_nLine = 0; g_nList = 4;</pre>	

2.改变颜色 void ShowSquare2(HDC hMemDc)

可知落地前为状态 1，令落地时为状态 2 (使用 void Change1To2())

```
int i = 0;
int j = 0;
HBRUSH hOldBrush;
HBRUSH hNewBrush = CreateSolidBrush( RGB(115,190,206));
hOldBrush = SelectObject(hMemDc,hNewBrush);
for(i = 0;i < 20;i++)
    for(j = 0;j < 20;j++)
        if(g_arrBackGround[i][j] == 2)    {
            Rectangle(hMemDc,  j*30, i*30, j*30+30, i*30+30); }
hNewBrush = SelectObject(hMemDc,hOldBrush);
DeleteObject;
```

3.成线消除 void DestroyOneLineSquare()

```
int i = 0;
int j = 0;
int nTempi = 0;
int nSum = 0;
for(i = 19;i >=0;i--)
{
    for(j = 0;j < 10;j++)
        nSum += g_arrBackGround[i][j];
    if(nSum == 20)
    { //消除一行并逐行下移
        for(nTempi = i - 1;nTempi >= 0;nTempi--)
            for(j = 0;j < 10;j++)
                g_arrBackGround[nTempi+1][j]=g_arrBackGround[nTempi][j];

        g_nSocre += 100; //计分
        i = 20; //i 回到最底下
        if(g_nSocre % 500 == 0 && (timeCounter >= 100))
        {
            g_nLevel ++;
            timeCounter -=50;
        }
    }
    nSum = 0; //清 nSum 为 0
}
```

4.旋转图形

采用逆时针旋转

a. 细长条形 void ChangeLineSquare()

b. 其他图形 void ChangeSquare()

正方形不需要做选择变换

判断能否旋转 int CanLineSquareChange()
int CanSquareChangeShape()

5.其他函数

void OnPaint(HDC hDc);

void PaintSquare(HDC hMemDc); //显示方块

产生随机块 int CreateRandSquare()

随机块贴进背景 void CopySquareToBack()

控制方块的移动 void OnCreate() void SquareDown()
void SquareLeft() void SquareRight()

定时器响应函数 void OnTimer(HWND hWnd)

ENTER 回车键 void OnReturn(HWND hWnd)

void OnLeft(HWND hWnd); void OnRight(HWND hWnd);

void OnDown(HWND hWnd); void OnChangeSquare(HWND hWnd);

未消除方块悬停 int CanSquareDown(); int CanSquareDown2();

判断能否左移 int CanSquareLeft(); int CanSquareLeft2();

判断能否右移 int CanSquareRight(); int CanSquareRight2();

判断游戏结束 int CanGameOver(HWND hWnd);

显示当前得分 void ShowScore(HDC hMemDc);

全部源代码将由附件展示。

四.心得体会

在这为期半个月的时间内，通过我们小组各成员之间的相互讨论和合作，在组长的带领下，我们有方向有计划地安排大作业的进程，完成了本次《俄罗斯方块》大作业的设计和优化。在这个过程中，本组的每个成员都根据自己的分工安排，在程序设计方面的学习和能力得到了充分的领悟和提高。这一次，通过我们自己的构思，不仅巩固了我们以前所学的知识，还对 C 语言有了更深一步的了解，掌握了更多的技巧和技能。在我们小组有解决不了的问题时，我们会主动在图书馆查阅相关的资料，或向其他老师和同学咨询，这个过程锻炼了我们动手实践解决问题的能力。复习 C 语言的过程中，不仅仅是课堂知识的回顾，更是对已有知识更加灵活的应用和创新。

很感谢这次的程序设计训练的课程学习，它使我们更加深刻地体会到对于理论知识的实践操作的重要性，只有掌握了一定量的专业知识才能得心应手地解决诸多问题；另外，做任何事都要有耐心，不要一遇到困难就退缩；在学习和工作中要时刻谨记“团结”二字，它好比通向成功的铺路石，不可或缺。