

山西大同大学

本科毕业论文设计)

题目：基于 C 程序的俄罗斯方块游戏设计

学院：物理与电子科学学院

## 基于 C 程序的俄罗斯方块游戏设计

**摘要：**俄罗斯方块是上世纪 70 年代设计出，是一个风靡全球的小简易液晶小游戏，虽然已经老旧，但是作为学习 c 语言还是可以作为一个很好的范例。俄罗斯方块中使用的方块是有四个小正方形小黑色块组成，建立一个结构体将方块存放在这个机构体中。如今技术的发展，可以为色块提供其他色彩，也可以增加色块的立体感，加大了游戏的美观性。游戏的使用着可以通过上下左右等键对方块移动进行控制。

俄罗斯方块通常有七种类型的方块，需要使用 c 程编写函数控制它的生成移动已经消除。玩游戏的人可以通过点击电脑键盘的控制键来控制方块的移动已经变形，经过一定的摆放，程序判断是否消除以及执行消除。俄罗斯方块要做出简单的图形，使用 vc++ 与 easyx 软件进行编译。文中将说明软件的使用。文中使用的 c 语言需要在 Windows 7 旗舰版下实现。

**关键词：**俄罗斯方块 ; easyx; C 语言

# 目 录

1 前言 .....	1
2 初步设想 .....	1
2.1 概述 .....	1
2.2 俄罗斯方块的历史与前景 .....	1
2.2 系统设计流程图 .....	2
3 游戏的运行环境 .....	3
3.1 硬件环境 .....	3
3.2 软件环境 .....	3
3.3 关于 <b>easyx</b> 的介绍 .....	4
4 游戏的具体设计 .....	4
4.1 方块的设计 .....	4
4.2 游戏界面的初始化 .....	6
4.3 游戏界面加入背景图片 .....	7
4.4 游戏加入背景音乐 .....	7
4.5 方块的初始化 .....	8
4.6 对方块的移动动态设计控制 .....	9
4.7 满行处理 .....	11
5 测试运行 .....	12
5.1 游戏代码输入到 <b>vc</b> 中 .....	12
5.2 游戏结束时弹出来的对话框 .....	12
5.3 绘图框中的游戏界面 .....	12
图 5 .....	13
结束语 .....	14
参考文献 .....	15
致谢 .....	15

## 1 前言

在这个科技发展迅猛的时代，电子类产品已经深入到我们每个人的生活当中，成为我们日常生活中不可缺少的一部分。俄罗斯方块游戏的出现可以说是游戏历史上的一件大事情，曾经创造的经济价值是不可估量的。该游戏曾经被全世界人们所追捧，我们中国人也不另外。小时候黑白机中游戏俄罗斯方块可以说是必不可少的。如今，俄罗斯方块仍然没有离开我们的实现，qq 游戏等诸多游戏平台以及网站都提供了俄罗斯方块的改良版，譬如双人游戏，积分游戏等。每每看到俄罗斯方块游戏都会勾起我们童年的一些回忆，可以说，俄罗斯方块已不再单单是一款游戏，它已经是我们回忆的一部分，深深的烙在我们心中。

俄罗斯方块游戏随着计算机技术已经电子技术的发展而不断进步，而且在俄罗斯方块游戏基础上也推出了各种游戏，比如说玛丽医生，相信这款游戏也得到很多人的青睐。俄罗斯方块游戏操作简单，它的原理其实就是运动的方块自上而下下落，在底部堆叠组合，并按规则消除。对于很多人而言，游戏的设计是高深莫测的，具有很高的专业技术难度。俄罗斯方块作为一个比较老但是很具有代表性的游戏，对于一些新手如我这样的人来，不失为一个学习的好游戏，而且，如今信息交流发达，可提供借鉴的材料也很多。本论文使用 `vc++6.0` 以及 `easyx` 对俄罗斯方块游戏代码进行编写，提供了一个学习 `c` 的机会，以便以后学习并开放更好的游戏或者应用软件。

该简易俄罗斯方块游戏是在当下用户最多的 `windows` 操作系统下运行，拥有常见俄罗斯方块的基本功能，游戏玩家可通过键盘控制游戏区域中的活动方块进行操作。设计的方块是随机出现的，并且拥有一定的色彩，提高了游戏的观赏性和可玩性。游戏方块在下落到底板后堆叠，满一行就消去这一行。值得注意的是本论文中加入了背景图片跟音乐，使这个游戏更具有趣味性。

## 2 初步设想

### 2.1 概述

综合自己在大学四年期间学习过的 `c` 语言知识，并通过自己查阅各种资料，设计俄罗斯方块游戏。通过设计游戏的过程，让自己明白游戏开发的过程，为将来走向社会打好基础。从游戏的设计到完成，提高自己的 `c` 语言编译能力。

### 2.2 俄罗斯方块的历史与前景

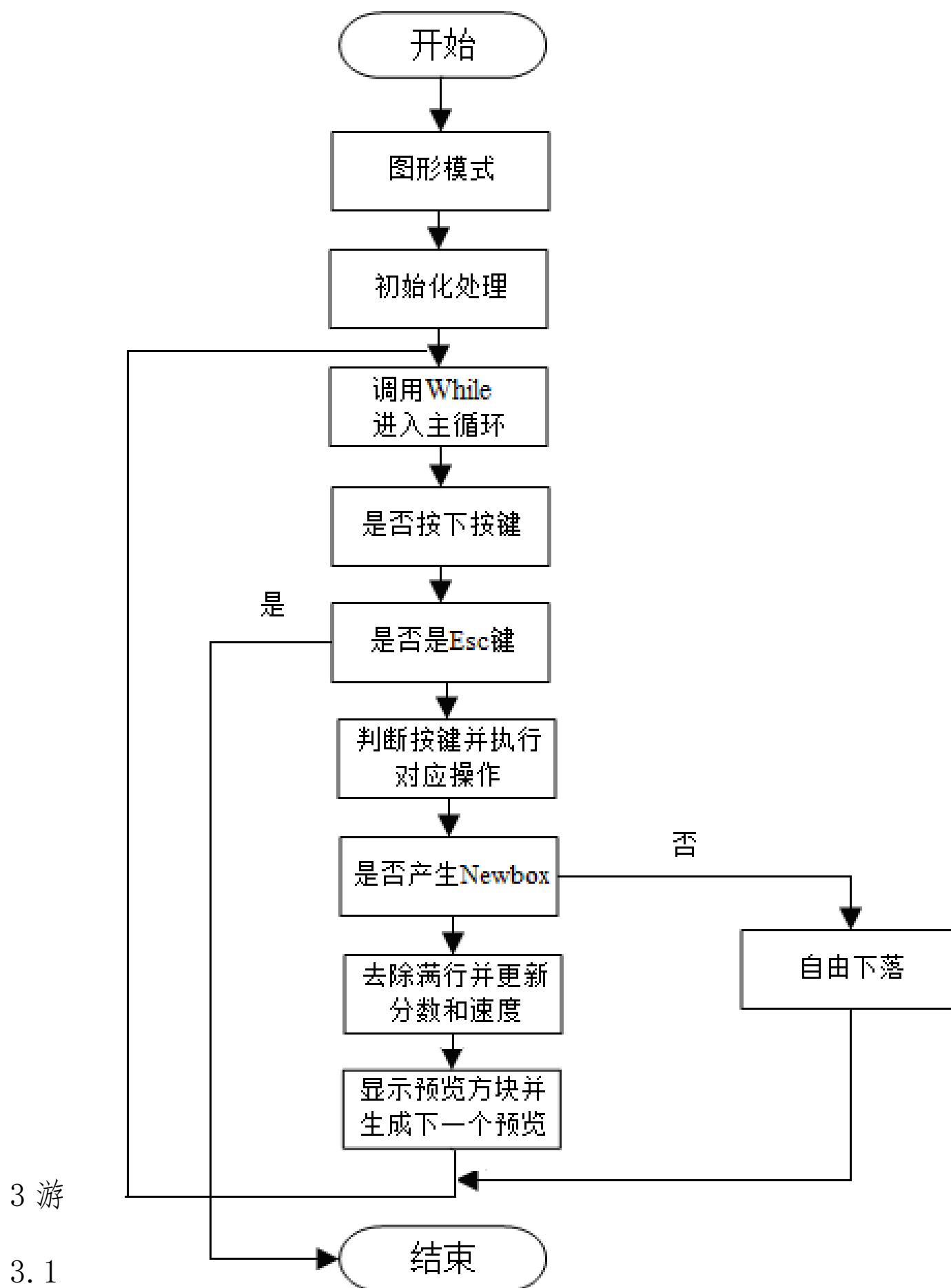
俄罗斯方块顾名思义，这个小游戏一定跟俄罗斯有关。在 1984 年的时候，

一个俄罗斯的小程序员受到了拼图游戏的启发，经过一段不为人知的研究，编写了俄罗斯方块游戏，他的名字叫阿列克谢·帕吉特诺夫。这个游戏在发明以后到现在，一直有很多人热爱。据说，俄罗斯方块游戏至少有 10 亿人玩过，这个数据是不完全统计。我认为，单单就中国这个拥有 13 亿劳苦大众的市场来说，这个游戏影响的不止一代人，而是好几代人。我相信俄罗斯方块游戏是有史以来，迄今为止使用人数最多的游戏，也是最火的游戏，30 多年来，俄罗斯方块游戏一直都存在在我们的事业中，你在 qq 游戏平台，以及别的平台，还有各个网页平台都能搜索到俄罗斯方块游戏，现代人给了俄罗斯方块更多的定义，可以有双人游戏，甚至还有了积分游戏，玩家可以通过游戏累加积分得到不同的更好的道具以增加游戏的趣味性。

在俄罗斯方块刚刚被作者创造出来的时候，因为俄罗斯还是一个社会主义国家，而且还处在冷战时期，游戏的开发已经都有一定的局限性，在当时的俄罗斯并没有推广开来，更没有形成一个产业。人们只是使用电脑复制游戏来玩，仅仅作为一个娱乐的小程序，并没有想到将来造成的经济影响。直到后来几年一个英国的软件公司的制造商到匈牙利拜访时看到有人玩彩色的俄罗斯方块游戏，这个商人以敏锐的经济眼光看到了这个游戏的发展前景，于是，他便到俄罗斯找到游戏的创始人商谈游戏的开发问题。再后来的几年，各大软件制造商开始了疯抢游戏的分销权。也是因为这种竞争，俄罗斯方块传到世界各地，当然也有中国这个大市场。从此，俄罗斯方块游戏影响了中国几代人。

虽然俄罗斯方块的发展经历了 30 年，但是，依然没有减退她的光辉。在现代电子技术的发展下，俄罗斯方块出现了 3D 模式，据说 2013 年还出现了《决战俄罗斯》。

## 2.2 系统设计流程图



本论文中的代码头在 windows 1 旗舰版下运行的，内存为 2G，配有键盘鼠标显示器，程序运行的最低硬件要求有待考究。

### 3.2 软件环境

本论文中的编写使用了 visual C++6.0 ， easyx2013 最新版。

easyx 为 vc++6.0 提供了图形库 graphics.h ，为游戏的设计提供了便利条件。在游戏设计不再单单的创建 file 文件这么简单，步骤如下：

1. 点击 file ，点击 projects ，创建 win32 console application ，定义工程名。注意不能加后缀。确定以后工程创建完毕。

2. 点击左侧界面内的工作区，点击已经创建好的工程，打开 source files 。
3. 点击 file ， 创建 c++source file ， 填写文件名， 注意文件名不能加.c



后缀。

图 2

### 3.3 关于 easyx 的介绍

在大一期间学习 c 语言是用到了 vc，vc 有较好的编辑和调试能力，但是却并没有 turbo c 的绘图能力，而且 turbo c 这个软件没有更新，过于老旧，连最基本的复制粘贴这种功能都没有。基于这种这种原因，还是用 vc 设计俄罗斯方块游戏，利用 easyx 提供的图形库设计俄罗斯方块游戏。

在 easyx 官网下载最新版压缩包，通过解压找到 setup.hta 文件，点击这个文件，再点安装库。安装过程中会自动检测到电脑已装的 vc，将库文件和头文件分别放入到 vc 对应的 include 和 lib 文件夹中。也可以手动操作，将文件复制到对应文件夹中，这样就可以删掉 easyx。

安装完毕后，输入范例程序运行，检查软件必要的环境是不是正确。

## 4 游戏的具体设计

### 4.1 方块的设计

首先是要设计游戏的方块，方块是有 4\*4 即 16 个小色快构成的，0 就是没

有色块，1 就是有色块。按照既定的规律，方块有 7 种，分别有 z 型，反 z 型，t 型，7 型，反 7 型，1 型，口型，以下代码为定义结构体 `boxs`，`boxs` 结构体中 7 个元素就是上述提到到 7 个类型。

```

boxs[7]={
/* □
 * □
 * □      □□□□
 * □
 */
      {0x0F00, 0x4444, 0x0F00, 0x4444, blue},
/*
 * □□
 * □□
 */
      {0x0660, 0x0660, 0x0660, 0x0660, red},
/*
 * □      □□□      □□      □
 * □      □      □      □□□
 * □□      □
 */
      {0x4460, 0x02E0, 0x0622, 0x0740, YELLOW},
/*
 * □      □□ □□□
 * □□      □      □
 * □□ □□□ □
 */
      {0x2260, 0x0E20, 0x0644, 0x0470, MAGENTA},
/*
 * □      □□
 * □□      □□
 * □

```



```

*/
    {0x0C60, 0x2640, 0x0C60, 0x2640, GREEN },
/*
*   □
*   □ □           □ □
*   □           □ □
*/
    {0x0360, 0x4620, 0x0360, 0x4620, CYAN},
/
*   □           □           □
*  □ □ □       □ □   □ □ □   □ □
*           □           □           □
*/
    {0x4E00, 0x4C40, 0x0E40, 0x4640, BROWN}};

```

## 4.2 游戏界面的初始化

`initgraph(640, 480)` 绘图的坐标体系范围是 `x640` 像素，`y480` 像素。坐标体系默认 `x` 轴右为正方向，`y` 轴是以下为正方向，这个是值得注意的，度量单位是像素。

坐标原点可以通过 `setorigin()` 函数修改游戏左下侧设置了游戏的操作说明，使用 `wasd` 操作方向。

```

// 设置坐标原点
setorigin(220, 20);
// 绘制游戏区边界
rectangle(0, 0, WID* SIZE, HEI * SIZE);
rectangle((WID + 1) * SIZE , 0, (WID + 5) * SIZE, 5 * SIZE);
//显示操作说明

```

宋体

操作说明

上：旋转

左：左移

右：右移

空格：沉底

：退出

#### 4.3

为了增加游戏的美观度，我为游戏背景增加了图片，图标路径用双斜杠。

```
// 主函数
```

```
void main()
```

```
{
```

```
    // 使用 640*480 色素的绘图框初始化
```

```
    initgraph(640, 480);
```

```
    // 找到图片，将图片引用到绘图框
```

```
    // 按任意键退出
```

```
    getch();
```

```
    closegraph();
```

```
}
```

#### 4.4 游戏加入背景音乐

为了增加游戏的娱乐性，我为游戏添加了背景音乐，音乐使用童年经典的叮当猫。加音乐的部分多使用了一个库文件，导入 `winmm.lib` 库，告诉编译器要用到得 `winmm.lib` 库，这样就可以使用 `windows` 的多媒体，并找到音乐开始播放。

此处的音乐文件要放到程序新建的工程文件夹中，即 `project` 中，`pragma` 可以搜索工程文件夹找到音乐文件，并利用 `windows` 多媒体的播放器将找到的音乐播放。

```
#include <easyx.h>
```

```
#include <conio.h>
```

```

void main()
{
    initgraph(640, 480);
    //
        叮当猫
        俄罗斯方块游戏

    _getch();
    //使用 windows 的播放器播放音乐

    _getch();
    closegraph();
}

```

#### 4.5

初始化俄罗斯方块，编写代码将方块表示在游戏中。用到了 **bar** 跟 **bar3d** 这两个函数，这两个函数同属于图形库。这两个函数是 **easyx** 新增的函数，是具有三维立体效果，大大的提高了游戏的可观赏性。一下仅介绍 **bar3d**（）的用法，该函数用于画有边框填充矩形。函数中有 **depth** 这个参数，矩形深度，使方块具有立体感。

```

void bar3d(
    int left,           矩形左坐标 x
    int top,            矩形上坐标 y
    int right,          矩形右坐标 x
    int bottom,         矩形下坐标 y
    int depth,          矩形的深度
    bool topflag        当该值为 false 时，不画顶部
);

```

具体代码如下

```

void Drawboxs(boxsINFO _boxs, DRAW _draw)
{
    WORD b = g_boxs[_boxs.id].dir[_boxs.dir];

```

```

int x, y;

int color = BLACK;
switch(_draw)
{
    case SHOW: color = g_boxs[_boxs.id].color; break;
    case HIDE: color = BLACK;      break;
    case FIX:
        BYTE r = GetRValue(g_boxs[_boxs.id].color) * 3 / 4;
        BYTE g = GetGValue(g_boxs[_boxs.id].color) * 3 / 4;
        BYTE b = GetBValue(g_boxs[_boxs.id].color) * 3 / 4;
        color = RGB(r, g, b);
        break;
}
setfillcolor(color);

for(int i=0; i<16; i++)
{
    if (b & 0x8000)
    {
        x = _block.x + i % 4;
        y = _block.y - i / 4;
        if (y < HEI)
        {
            if (_draw != HIDE)
                bar3d(x * SIZE + 2, (HEI - y - 1) * SIZE + 2, (x + 1)
* SIZE - 4, (HEI - y) * SIZE - 4, 3, true);
            else
                bar(x * SIZE, (HEI - y - 1) * SIZE, (x + 1) * SIZE - 1,
(HEI - y) * SIZE - 1);
        }
    }
    b <<= 1;
}
}

```

#### 4.6

解释程序中对方块动态控制的操作。

旋转操作中，规定顺时针旋转方向是正方向，由一个操作键控制，没有反方

```

void OnRotate()
{
    // 获取可以旋转的 x 偏移量
    int dx;
    BLOCKINFO tmp = g_CurBoxs;
    tmp.dir++;          if (Checkboxs(tmp))    { dx = 0;
    goto rotate;      }
    tmp.x = g_Curboxs.x - 1;  if (Checkboxs(tmp))    { dx = -1;    goto
rotate;      }
    tmp.x = g_Curboxs.x + 1;  if (Checkboxs(tmp))    { dx = 1;
    goto rotate;      }
    tmp.x = g_Curboxs.x - 2;  if (Checkboxs(tmp))    { dx = -2;    goto
rotate;      }
    tmp.x = g_Curboxs.x + 2;  if (Checkboxs(tmp))    { dx = 2;
    goto rotate;      }
    return;

rotate:
    // 旋转
    Drawboxs(g_boxs, HIDE);
    g_Curboxs.dir++;
    g_Curboxs.x += dx;
    Drawboxs(g_CurBlock);
}

```

左移操作，只要把将方块的横坐标加一减一即可。左减右加。

// 左移方块

```

void OnLeft()
{
    boxsINFO tmp = g_Curboxs;
    tmp.x--;
    if (Checkboxs(tmp))
    {
        Drawboxs(g_Curboxs, HIDE);
        g_Curboxs.x--;
    }
}

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/825301104334011221>