**方块拼图课程设计报告**

学 号 13080005 .

姓 名 周 源 .

指导教师 陈 彩 .

2013年12月

# 目录

[1 需求分析 4](#_Toc375683668)

[1.1 功能与数据需求 4](#_Toc375683669)

[1.1.1欢迎 4](#_Toc375683670)

[1.1.2登录游戏 4](#_Toc375683671)

[1.1.3初始游戏 4](#_Toc375683672)

[1.1.4判断点击游戏界面 4](#_Toc375683673)

[1.1.5画出临时棋子 4](#_Toc375683674)

[1.1.6画出棋子 4](#_Toc375683675)

[1.1.7消除棋子 5](#_Toc375683676)

[1.1.8游戏结束 5](#_Toc375683677)

[1.1.9拓展：悔棋 5](#_Toc375683678)

[1.1.10显示排行 5](#_Toc375683679)

[1.1.11显示帮助 5](#_Toc375683680)

[1.1.12显示设置 5](#_Toc375683681)

[1.1.13修改/添加/删除用户 5](#_Toc375683682)

[1.1.14拓展：修改棋子颜色 5](#_Toc375683683)

[1.1.15拓展：商店 6](#_Toc375683684)

[1.2界面需求 6](#_Toc375683685)

[1.2.1欢迎界面 6](#_Toc375683686)

[1.2.2游戏界面 6](#_Toc375683687)

[1.2.3大消息框 6](#_Toc375683688)

[1.2.4小消息框 6](#_Toc375683689)

[1.3开发与运行环境需求 6](#_Toc375683690)

[1.4 其他方面需求 6](#_Toc375683691)

[2 概要设计 7](#_Toc375683692)

[2.1 主要数据结构 7](#_Toc375683693)

[2.2 程序总体结构 7](#_Toc375683694)

[2.2.1管理模块 7](#_Toc375683695)

[2.2.2存储模块（Save系列函数） 7](#_Toc375683696)

[2.2.3点击控制模块（Click系列函数） 8](#_Toc375683697)

[2.2.4棋子控制模块（Block系列函数） 9](#_Toc375683698)

[2.2.5显示模块（Show系列函数） 10](#_Toc375683699)

[3 详细设计 11](#_Toc375683700)

[3.1 管理模块 11](#_Toc375683701)

[3.2 存储模块 11](#_Toc375683702)

[3.3 点击控制模块 12](#_Toc375683703)

[3.4 棋子控制模块 13](#_Toc375683704)

[3.4 显示模块 13](#_Toc375683705)

[4 测试 14](#_Toc375683706)

[5 用户手册 17](#_Toc375683707)

[5.1 程序功能 17](#_Toc375683708)

[5.2 运行环境要求 17](#_Toc375683709)

[5.3 安装与启动方法 18](#_Toc375683710)

[5.4 程序的界面、交互方式和操作方法 18](#_Toc375683711)

[5.5 输入数据类型、格式和内容限制 18](#_Toc375683712)

[5.6 用户需要使用的交互命令名称、功能和格式 18](#_Toc375683713)

[6 总结提高 18](#_Toc375683714)

[6.1 课程设计总结 18](#_Toc375683715)

[6.2 对本课程意见与建议 18](#_Toc375683716)

[附录 19](#_Toc375683717)

[附录1 方块拼图游戏更新历史 19](#_Toc375683718)

[附录2 程序源代码 21](#_Toc375683719)

[附录2.1 main.cpp 21](#_Toc375683720)

[附录2.2 Save.h 30](#_Toc375683721)

[附录2.3 Click.h 37](#_Toc375683722)

[附录2.4 Block.h 46](#_Toc375683723)

# 1 需求分析

本课程设计旨在完成名为“方块拼图”的游戏，其功能大致为：在棋盘上放置7种不同的棋子，通过形成整块面积的棋子消除得分。在游戏的基础上添加了排名、帮助、设置等功能。

## 功能与数据需求

### 1.1.1欢迎

打开游戏后，如果用户之前没有存档，则只需点击屏幕中的按钮或者用键盘按Enter键即可触发登录系统；如果有存档，游戏将主动提示是否读取存档。

### 1.1.2登录游戏

每个玩家都有一个账号，当游戏开始时需要登录。如果在输入框输入已经存在的用户，游戏将判断该用户是否有密码，如果有则会要求输入密码。如果用户输入密码错误五次游戏将返回到输入用户名界面。如果输入框中输入了不存在的用户，游戏将自动创建新用户，并且询问请求设置密码，用户可以选择不设置，设置密码或输入正确密码后将进入游戏界面。

### 1.1.3初始游戏

游戏界面函数功能包括：游戏开始时和过关时初始化游戏界面、如果有存档读取数据、显示下一个棋子、判断是否能够继续游戏、对每次点击游戏界面操作进行处理。

### 1.1.4判断点击游戏界面

此功能会对鼠标或键盘对于在游戏界面做出的点击进行判断，当鼠标点击中央棋盘区域、各种功能按钮，或者键盘点击方向键、各种替代功能按钮的按键时，该功能函数返回特定值。

### 1.1.5画出临时棋子

当鼠标移动到中央棋盘区或者键盘按下方向键后，游戏将在棋盘上显示出棋子放置在此处的临时状态。再次功能中棋子的颜色为正常棋子颜色亮度加0.5对应的浅色棋子。如果该处不能放置，则临时棋子的样式为灰色棋子上有一个红色叉子。

### 1.1.6画出棋子

当鼠标点击中央棋盘区域或者键盘按下回车键，且该处可以放置棋子时，游戏将在该处放置棋子。

### 1.1.7消除棋子

此功能每放置一个棋子后执行一次，当屏幕中出现面积大于等于3×3面积的棋子时，游戏将消除这些棋子，并且在总分中加入相应分数。其中，如果游戏局面中出现同时出现两块可消除的区域但有部分重合时，游戏将优先消除面积大的部分；如果面积相等，将优先消除左上角行列值小的区域；如果左上角棋子为同一个将优先消除高度大于宽度的区域。

### 1.1.8游戏结束

此功能每放置一个棋子后执行一次，当整个棋盘都不能再放下即将出现的棋子时，游戏结束。此时用户将看到游戏结束后的排行，并且可以选择重新游戏或者返回欢迎界面。

### 1.1.9拓展：悔棋

此功能基于在每放置一个棋子后，游戏将把每次操作的信息存储在文件中。当玩家在已经放置了棋子以后可以通过悔棋功能回到之前的状态。

### 1.1.10显示排行

该功能将在点击“排名”按钮时执行。游戏将首先从分数保存文件中读取所有用户的分数记录形成链表，然后将前十名显示在界面上。最后从开始查找当前用户的排名。

### 1.1.11显示帮助

该功能将在点击“帮助”按钮时执行。游戏将依次载入帮助图片文件中的内容，形成动画。同时在动画过程中可以随时点击关闭帮助界面。

### 1.1.12显示设置

该功能将在点击“设置”按钮时执行。该界面中可以选择修改用户名、修改密码、新建用户、删除用户、修改难度以及修改棋子颜色。

### 1.1.13修改/添加/删除用户

这一系列功能都会从存储用户列表的文件中形成链表，然后对链表进行修改，最后再重新写入文件中。

### 1.1.14拓展：修改棋子颜色

该功能将可以修改棋子颜色，在界面中会显示一个调色板，当用户点击某颜色后，函数返回相应的坐标值代替其颜色。

### 1.1.15拓展：商店

在商店中有多种不同功能的道具，通过消耗相应的分数（金币）兑换。功能包括将所有灰色棋子删除、将所有灰色棋子变为可以消除的棋子、任选棋子样式、使用只占一格的棋子、消除一个非灰色棋子、消除3×3面积内的任意棋子等。

## 1.2界面需求

### 1.2.1欢迎界面

载入欢迎界面图片，以及开始按钮。

### 1.2.2游戏界面

游戏界面上方为标题以及功能按钮，右侧上方显示游戏当前信息，包括等级、分数（金币）和距离过关还需要放置的棋子数量，右侧下方显示即将出现的棋子形状。游戏界面主题为一8×8棋盘。

### 1.2.3大消息框

大消息框用于显示帮助、设置以及排行等功能。载入大消息框图片。

### 1.2.4小消息框

小消息框用于提示、警告、询问用户信息。载入小消息框图片。

## 1.3开发与运行环境需求

开发环境：Microsoft Visual C++ 6.0。运行环境：Windows XP/Vista/7/8.x。

## 1.4 其他方面需求

无

# 2 概要设计

## 2.1 主要数据结构

玩家信息结构体：用于存储玩家在游戏时的当前基本信息，具体包括关卡、分数、剩余棋子数、难度、可悔棋次数、棋子颜色等。

用户链表：用于存储所有玩家的名称、密码和分数。

游戏现状链表：由于存储当前玩家所有操作的游戏信息。

## 程序总体结构

### 2.2.1管理模块

功能：对游戏整体进行操控。

函数：

1. void main()

功能：对模块调用进行总控制。

参数：无。

返回值：无。

1. int Game(char player[], bool read)

功能：对游戏过程进行处理。

参数：char player[]：游戏玩家名称；bool read：是否要读取未完成的游戏。

返回值：需要回到欢迎界面返回1，需要退出返回0。

### 2.2.2存储模块（Save系列函数）

功能：便于用户存储、读取游戏记录。

函数：

1. bool SaveLogIn(char player[])

功能：登录主函数。

参数：char player[]：用于存储玩家名称。

返回值：成功登录返回1，取消登录返回2。

1. bool SaveChange(int type, char player[])

功能：更改存储信息主函数。

参数：int type：更改类型（1为更改用户名，2为更改密码）；char player[]：玩家名称。

返回值：成功更改返回1，取消更改返回2。

1. bool SaveCreate()

功能：添加用户的主函数。

参数：无。

返回值：成功更改返回1，取消更改返回2。

1. bool SaveDelete(char player[])

功能：删除用户的主函数。

参数：char player[]：当前玩家用户名（可以删除自己）。

返回值：成功更改返回1，取消更改返回2。

1. void SaveScore(int score, char player[])

功能：保存用户分数主函数。

参数：int score：游戏结束时的分数；char player[]：当前玩家用户名。

返回值：无。

1. void SaveBack(bool Has[], bool Grey[], INFO \*nf, int \*NewBlock, char player[])

功能：读取悔棋存档的主函数。

参数： bool Has[]：棋子放置情况；

bool Grey[]：灰色棋子放置情况；

INFO \*nf：玩家当前所有数据信息；

int \*NewBlock：当前棋子样式；

char player[]：当前玩家用户名。

返回值：无。

1. void SaveRead(user \*H)

功能：读取用户列表（list.save文件）并存为链表的功能函数。

参数： user \*H：用户链表的头结点。

返回值：无。

1. bool SaveReadClose()

功能：读取存储列表（close.save文件）并存为链表的功能函数。

参数：无。

返回值：载入游戏记录返回1，不载入返回0。

1. bool SaveInputPass(user \*p, user \*input)

功能：输入密码的功能函数。

参数： user \*p：用户链表的一个结点，user \*input：输入用户的名称、输入密码。

返回值：没有密码或输入成功返回1，输入五次错误返回0。

1. void SaveNew(user \*input)

功能：保存新用户的功能函数。

参数： user \*input：输入用户的名称、输入密码。

返回值：无。

1. void SaveNodeDel(user \*H)

功能：删除链表的功能函数。

参数： user \*H：用户链表的头结点。

返回值：无。

### 2.2.3点击控制模块（Click系列函数）

功能：在不同界面中对鼠标或键盘操作进行实时监控（部分函数除监控外还对操作信息进行处理）。

函数：

1. bool ClickWelcome()

功能：**在欢迎界面**监测鼠标或键盘是否对界面有任何操作。

参数：无。

返回值：点击继续/Enter按钮返回1，点击Esc返回0。

1. void ClickMSGRank()

功能：**在显示排名界面**监测鼠标或键盘是否对界面有任何操作。

参数：无。

返回值：无。

1. int ClickMSGSet(int dif)

功能：**在设置界面**监测鼠标或键盘是否对界面有任何操作。

参数：int dif：当前游戏难度。

返回值：点击关闭/Enter/Esc按钮返回1，点击不同设置返回101~108。

1. int ClickMSGTool()

功能：**在商店界面**监测鼠标或键盘是否对界面有任何操作。

参数：无。

返回值：点击关闭/Enter/Esc按钮返回1，点击不同商品返回111~116。

1. int ClickToolFlash(int type, int time)

功能：在商店界面，当鼠标移动至某商品上时播放动画，同时可以监测操作。

参数：int type：商品序号；int time：每张图片播放时间。

返回值：点击该商品返回1，鼠标移出该商品区域返回0，未操作返回-1。

1. int ClickMSGRGB()

功能：**在设置棋子颜色界面**监测鼠标或键盘是否对界面有任何操作。

参数：无。

返回值：点击关闭/Enter/Esc按钮返回1，点击某种颜色返回其RGB值。

1. bool ClickMSGSml(char title[], char str[], int type)

功能：**在任意小消息框界面**监测鼠标或键盘是否对界面有任何操作。

参数：char title[]：提示框标题；char str[]：提示内容；int type：提示类型。

返回值：类型1、2（提示/错误）：点击关闭/Enter/Esc按钮返回0

类型3（选择）：点击是/Y返回1，点击否/N返回0。

1. bool ClickInput(char title[], char input[], char help[], bool pass)

功能：**在任意输入框界面**监测鼠标或键盘是否对界面有任何操作。

参数： char title[]：输入框标题；char input[]：输入内容；

char help[]：输入内容提示；bool pass：是否将输入内容显示为星号。

返回值：点击确定/Enter返回1；点击关闭/Esc按钮返回0。

1. void ClickMosaic()

功能：**在消息框或输入框界面**将背景马赛克化。

参数：无。

返回值：无。

### 2.2.4棋子控制模块（Block系列函数）

功能：控制棋子的显示、放置与消除等功能。

函数：

1. void BlockTemp(bool Has[], int RGBC, int RGBC\_, int NewBlock, int TypeBlock, int co)

功能：鼠标移动到棋盘时显示出当前棋子若放置后的效果预览。

参数： bool Has[]：棋子放置情况；

int RGBC：当前棋子颜色；

int RGBC\_：修改颜色前棋子颜色；

int NewBlock：当前棋子样式；

int TypeBlock：当前棋子方向；

int co：鼠标当前坐标。

返回值：无。

1. bool BlockDraw(bool Has[], bool Grey[], INFO nf, int NewNextB, int TypeBlock, int co)

功能：鼠标在棋盘上点击时放置棋子。

参数： bool Has[]：棋子放置情况；

bool Grey[]：灰色棋子放置情况；

INFO nf：当前玩家所有数据信息；

int NewNextB：要显示的棋子样式（当前棋子或下一个）

int TypeBlock：当前棋子方向

int co：鼠标当前坐标（值为0表示要显示下一个棋子）。

返回值：成功放置返回1，放置失败返回0。

1. void BlockDel(bool Has[], bool Grey[], int \*score)

功能：监测棋盘上的棋子是否可以消除。

参数： bool Has[]：棋子放置情况；

bool Grey[]：灰色棋子放置情况；

int \*score：游戏当前分数。

返回值：无。

1. int BlockCanPut(bool Has[], bool Grey[], int NewBlock)

功能：监测游戏是否可以继续进行。

参数： bool Has[]：棋子放置情况；

bool Grey[]：灰色棋子放置情况；

int NewBlock：当前棋子样式。

返回值：不能放置返回0，可以放置返回第一个可以放置棋子位置的坐标。

### 2.2.5显示模块（Show系列函数）

1. void ShowRank(char player[])

功能：显示排行的主函数。

参数：char player[]：当前用户名。

返回值：无。

1. void ShowHelp()

功能：显示帮助的主函数。

参数：无。

返回值：无。

1. bool ShowHelpTextFlash(char help[])

功能：显示帮助中显示文字的动画函数。

参数：char help[]：要显示的文字内容。

返回值：关闭帮助返回1，无操作返回0。

1. bool ShowHelpSleep(int time)

功能：显示帮助时动画每帧之间插入间隔时间（此部分与Sleep()函数功能相同），同时监测是否有鼠标或键盘操作的函数。

参数：int time：间隔时间。

返回值：有操作返回1，无操作返回0。

# 3 详细设计

## 3.1 管理模块

1. void main()见图1.1。
2. int Game(char player[], bool read) 见图1.2。

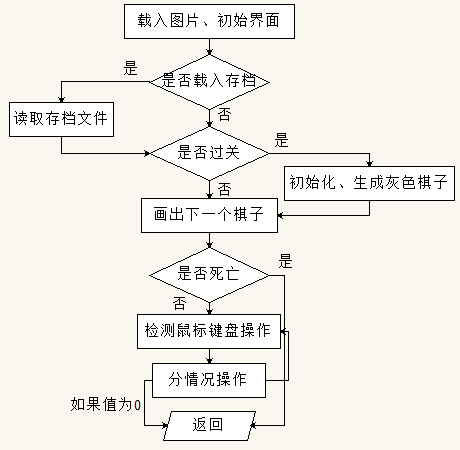
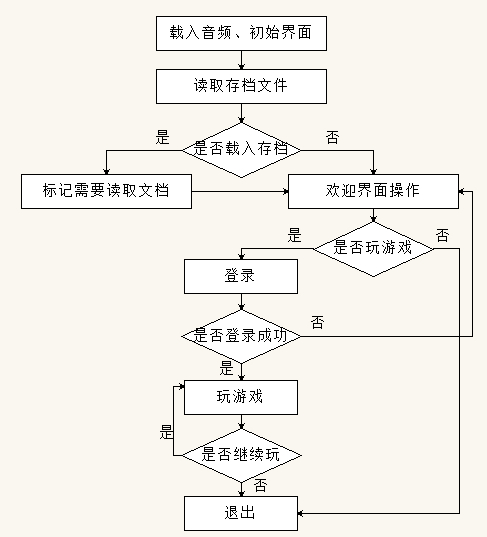


图1.1 图1.2

## 3.2 存储模块

1. bool SaveLogIn(char player[])见图2.1。
2. bool SaveChange(int type, char player[])见图2.2。

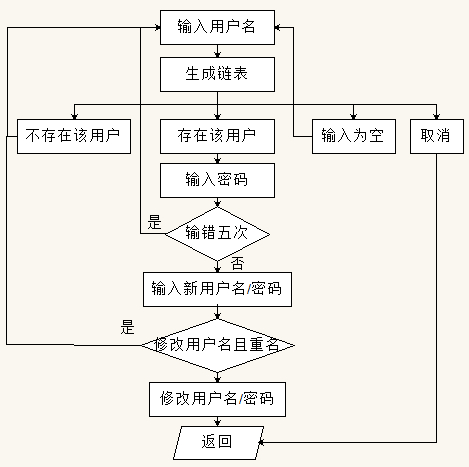
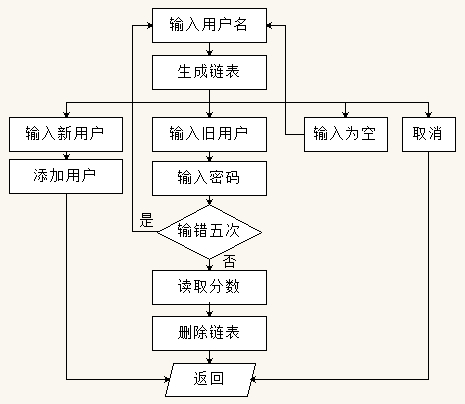


图2.1 图2.2

1. bool SaveCreate()见图2.3。
2. bool SaveDelete(char player[])见图2.4。
3. void SaveScore(int score, char player[])见图2.5。

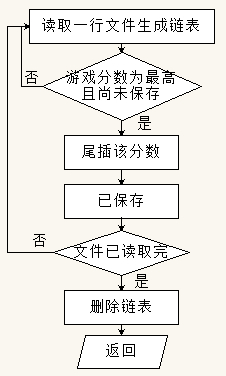
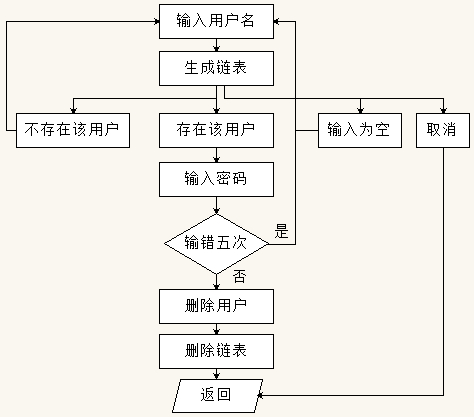
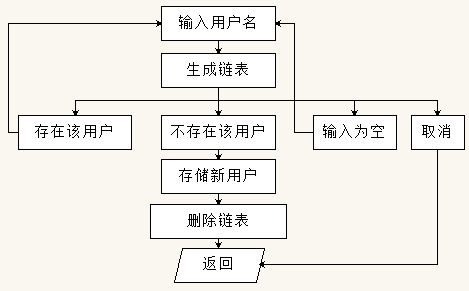


图2.3 图2.4 图2.5

1. void SaveBack(bool Has[], bool Grey[], INFO \*nf, int \*NewBlock, char player[])

流程：生成链表→读取最后一行→更改数据信息→将除去最后一行的链表存回文件→删除链表。

1. void SaveRead(user \*H)

流程：打开文档→读取每一行文件生成链表→关闭文件。

1. bool SaveReadClose()

流程：打开缓存文档→如果存在询问是否读取→关闭文件→返回是否读取。

1. bool SaveInputPass(user \*p, user \*input)

流程：如果有密码继续，否则直接返回→输入密码→如果输对则返回，否则重新输入→如果输错5次返回。

1. void SaveNew(user \*input)

流程：打开名单文档→设置密码→如果有密码则存入密码，否则存入0→关闭文件。

1. void SaveNodeDel(user \*H)

流程：创建删除结构体指针指向循环变量→循环变量步进→删除删除结构体所指空间。

## 3.3 点击控制模块

1. bool ClickWelcome()

流程：循环判断是否有鼠标或键盘点击→如果有返回0或1，否则继续循环。

1. void ClickMSGRank()

流程：循环判断是否有鼠标或键盘点击→如果有则结束，否则继续循环。

1. int ClickMSGSet(int dif)

流程：循环判断是否有鼠标或键盘点击→如果有返回1或101~108，否则继续循环。

1. int ClickMSGTool()

流程：循环判断是否有鼠标或键盘点击→如果有返回1或111~116，否则继续循环。

1. int ClickToolFlash(int type, int time)

流程：循环判断是否有鼠标或键盘点击→如果有返回0或1，否则继续循环。

1. int ClickMSGRGB()

流程：循环判断是否有鼠标或键盘点击→如果有返回1或RGB值，否则继续循环。

1. bool ClickMSGSml(char title[], char str[], int type)

流程：循环判断是否有鼠标或键盘点击→如果有返回0或1，否则继续循环。

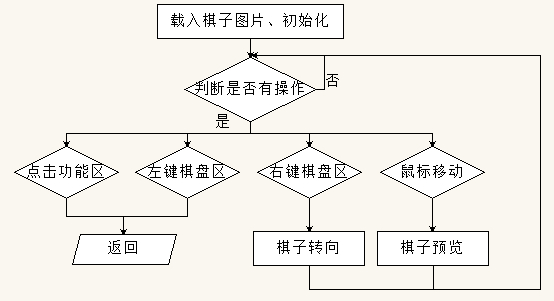
1. bool ClickInput(char title[], char input[], char help[], bool pass)

流程：循环判断是否有鼠标或键盘点击→如果有返回0或1，否则继续循环。

1. void ClickMosaic()

流程：获得背景图片→每15×15像素为一单位求RGB颜色平均值并存入→显示存入的图片。

int ClickPlay(bool Has[], bool Grey[], INFO nf, int NewBlock, char player[], bool \*wheel)见图2.6。

图2.6

## 3.4 棋子控制模块

1. void BlockTemp(bool Has[], int RGBC, int RGBC\_, int NewBlock, int TypeBlock, int co)

流程：判断棋子类型→如果可放置预览浅颜色的棋子，如果不可放预览带红叉的灰色棋子。

1. bool BlockDraw(bool Has[], bool Grey[], INFO nf, int NewNextB, int TypeBlock, int co)

流程：判断棋子类型与是否可放置→如果可放置则放置，否则提示不可放置。

1. void BlockDel(bool Has[], bool Grey[], int \*score)

流程：从左上角寻找可消除的棋子区域→如果存在区域则消除**并重新寻找**，如果不存在则返回。

1. int BlockCanPut(bool Has[], bool Grey[], int NewBlock)

流程：从左上角寻找可以放置棋子的区域→如果存在则返回坐标，否则返回0。

## 3.4 显示模块

1. void ShowRank(char player[])

流程：在屏幕载入一消息框→读取分数文件生成链表→输入前十名→重新查找当前玩家排名。

1. void ShowHelp()

流程：在屏幕载入一消息框→依次读取帮助图片中的内容显示在屏幕上→全部读取完毕后循环标题直到有操作使之结束。

1. bool ShowHelpTextFlash(char help[])

流程：在输入区域覆盖一白色框→按照每个字显示在屏幕上→间隔一段时间→结束。

1. bool ShowHelpSleep(int time)

流程：取开始、结束时间→循环判断时间差是否达到要求→如果没有监测是否有鼠标或键盘操作，否则退出循环→再取结束时间。

# 4 测试

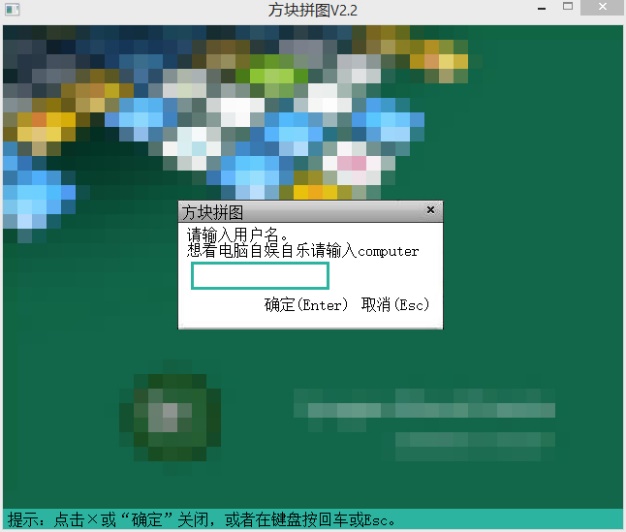
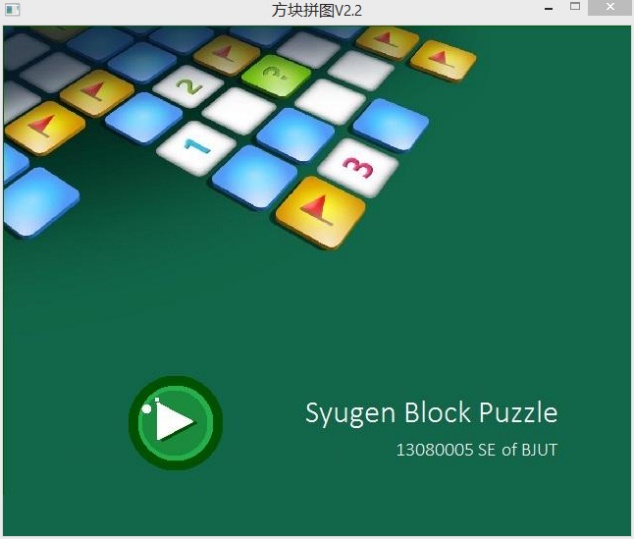


图4.1 开始界面 图4.2 输入用户名

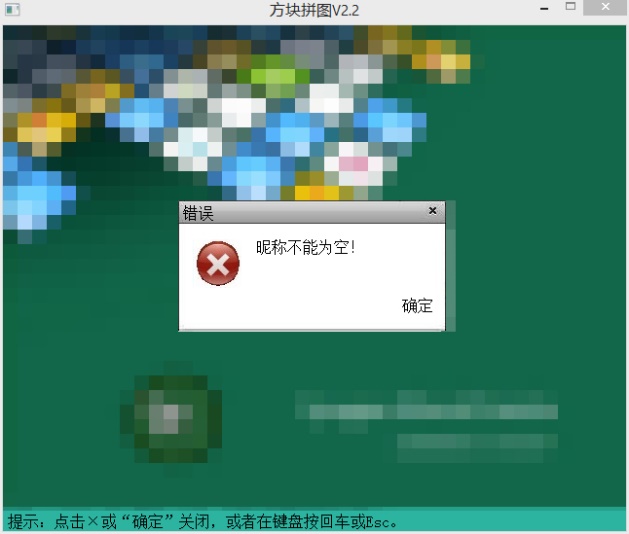


图4.3 没有输入昵称 图4.4输入密码



图4.5密码输入错误 图4.6密码五次输入错误



图4.7 登录成功 图4.8游戏界面 棋子预览

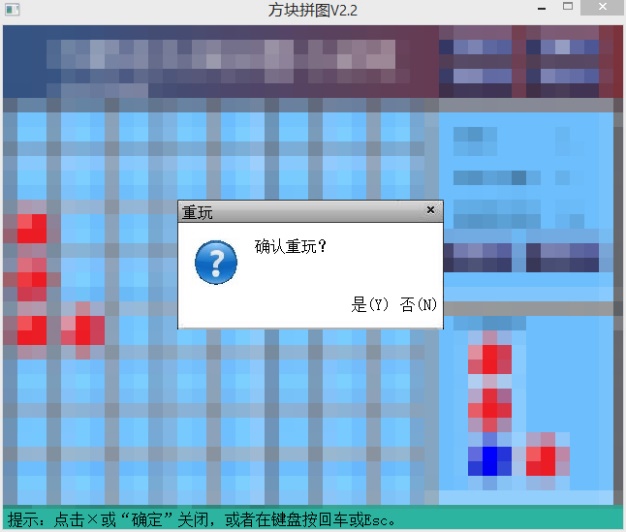


图4.9放置棋子 图4.10重玩

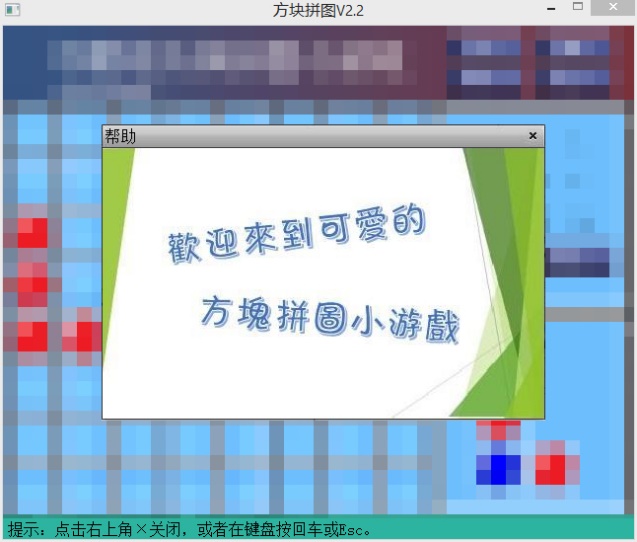


图4.11帮助 图4.12排名

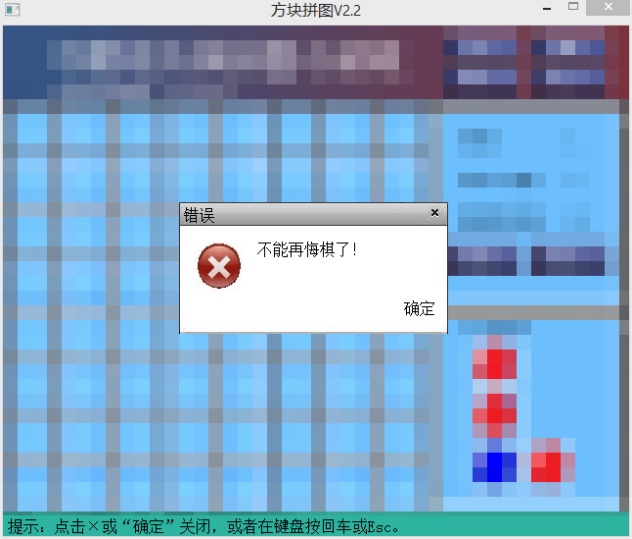
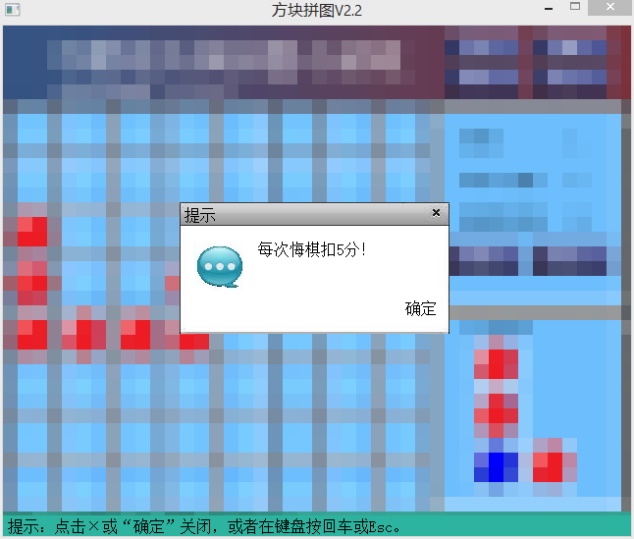


图4.13悔棋 图4.14没有棋子不能悔棋

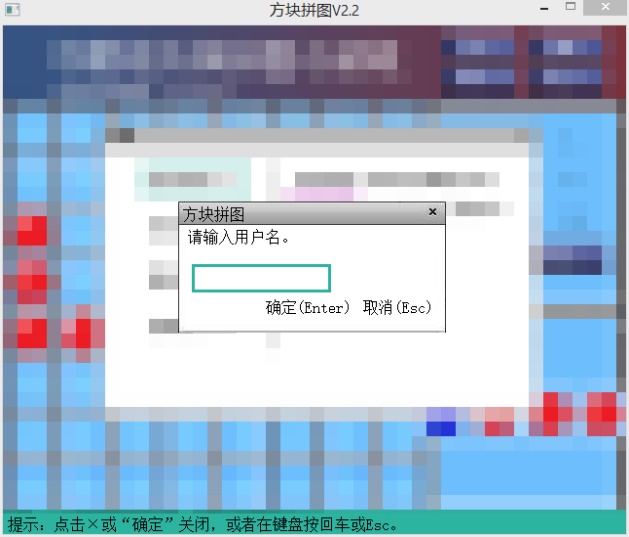
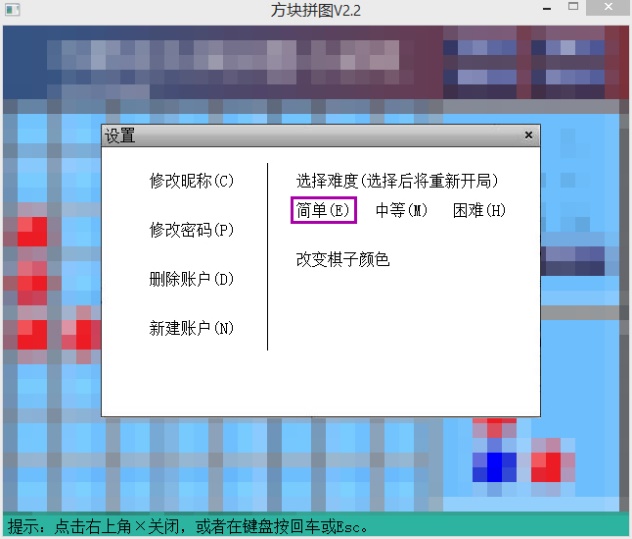


图4.15 设置 图4.16修改用户名



图4.17输入有误（原用户名不存在或新用户名已存在） 图4.18 修改成功

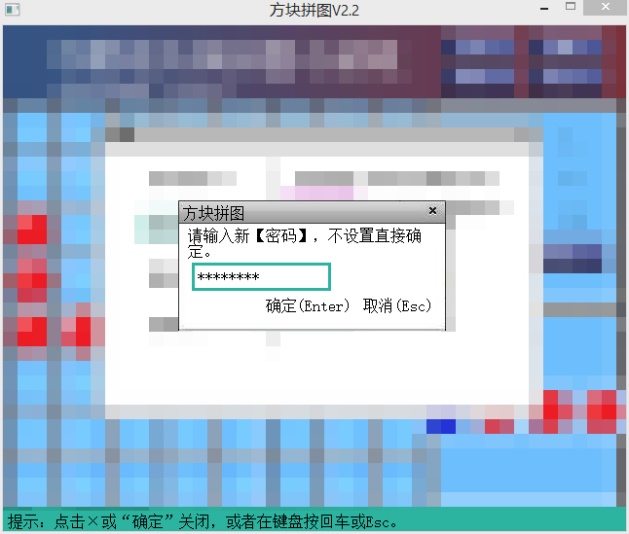


图4.19修改密码 图4.20修改难度

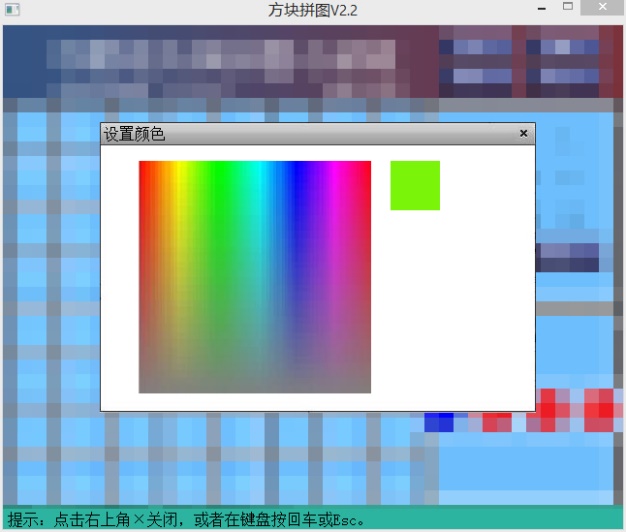


图4.21修改棋子颜色 图4.22购买道具

# 5 用户手册

## 5.1 程序功能

本程序为方块拼图游戏，玩家可以在棋盘上放置7种不同的棋子，通过形成整块面积的棋子消除得分。在游戏的基础上添加了排名、帮助、悔棋、购买道具、修改用户名称或密码、添加或删除账户、更改游戏难度、选择棋子颜色等功能。

## 5.2 运行环境要求

该游戏运行环境为Windows XP/Vista/7/8.x。

## 5.3 安装与启动方法

程序无需安装，直接双击文件“方块拼图Vx.x.exe”运行。

## 5.4 程序的界面、交互方式和操作方法

程序界面为一Windows窗口，通过鼠标与键盘进行点击操作与计算机进行交互。

## 5.5 输入数据类型、格式和内容限制

所有输入数据皆以字符串形式输入，**在输入时请注意使用英文输入法，否则输入内容无效**。

## 5.6 用户需要使用的交互命令名称、功能和格式

用户只需按照游戏中提示进行选择即可完成交互。

# 6 总结提高

## 6.1 课程设计总结

高级语言程序设计是我目前认为最吸引我的一门学科之一，原因在于我比较热爱编程。由于我在高一是自学过C++，而且在高二的时候用程序控制台编写过一个扫雷游戏，让我更加觉得编程的乐趣，因此这次的课设我带着期待的心情开始完成。

但实际上没有一个应用程序是很容易开发的，我在这次的方块拼图课设中也经历了许多困难，比如说为了完成一个拓展的功能，即更改棋子的颜色。由于我的棋子是载入的图片，而不是直接画图，因此setfillcolor函数并不能发挥作用。因此我登录了EasyX网站希望寻找解决方法，我发现有人发布过一个将图片打马赛克的函数，于是便借此学习了一番，发现可以更改存储在IMAGE型变量里的每一个像素的RGB值。而如何让用户选择RGB值成了另一个难题，最终我利用鼠标坐标先转化为HSL值（其中的H和S恰可以对应鼠标的x与y），再用HSLtoRGB函数转为RGB，由此便可以更改棋子的颜色。

在从11月14日开始对界面开始设计到如今，断断续续地改进程序中，我学到的不仅仅如上面提到的这一点点。总之，在大一的这次课设让我对C语言的应用能力变得更强，相信对今后的编程道路有着不可或缺的作用。

## 6.2 对本课程意见与建议

我认为这门课从整体上讲非常好，在短短的一学期内能让很多编程零基础的同学能达到自己也可以编写出优秀的程序的水平。如果希望提出一点建议，可能是希望老师能多提供一些针对当堂课的练习题，作为不需要提交的补充练习供有余力的同学加强知识。这是因为在我们班的讲课速度对于基础不是很好而又想取得好成绩的同学来说，想完全吸收课上所将知识有些困难，因此可以通过实例来巩固所学知识。

# 附录

## 附录1 方块拼图游戏更新历史

Beta版：（不具有游戏功能）

2013年11月14日：方块拼图Beta1.0.1114

1. 制作游戏界面（欢迎界面不能键盘操作）
2. **鼠标或键盘控制格子选定位置**

2013年11月15日：方块拼图Beta2.0.1115

1. 添加用户登录模块
2. 添加设置功能，包括更改昵称与密码、添加删除账号
3. 添加简单的帮助对话框

2013年11月17日：方块拼图Beta3.0.1117

1. 添加排名显示功能（bug只显示每个玩家的一个记录）

2013年11月19日：方块拼图Beta4.0.1119

1. 添加放置7种棋子功能，即可以正式进行游戏（不能旋转）
2. 添加消除3×3面积的棋子功能
3. 添加关卡概念（过关后没有清空画面）
4. **添加重玩功能**

2013年11月20日：方块拼图Beta5.0.1120

1. 添加旋转棋子功能
2. **添加判断游戏失败的功能**
3. **改进消除功能，可以消除任意3×3及以上面积（bug）**
4. 添加过关后清空屏幕功能
5. 添加关卡上限
6. 添加随机放置障碍棋子功能
7. 改进重玩功能，添加确认提示框（bug：取消后棋子会改变）

正式版：（可以进行游戏）

2013年11月23日：方块拼图1.0.1123

1. **添加显示目前玩家名称**
2. **添加排名中显示个人排名功能**
3. 解决了关闭设置/帮助/排名后棋子会改变的bug

2013年11月23日：方块拼图1.0b.1123

1. 解决了以L型和反L型棋子不能判断游戏结束的bug

2013年11月24日：方块拼图1.1.1124

1. **将系统消息框(MessageBox)改为自编消息框函数**
2. 添加选择游戏难度功能
3. **添加选择棋子颜色功能**
4. **添加出现消息框时背景打马赛克功能**

2013年11月25日：方块拼图1.2.1125

1. **添加悔棋功能**
2. **添加关闭游戏时提示是否保存游戏记录功能**
3. **添加游戏开始时提示是否载入游戏记录功能**
4. **添加消除棋子时的动画**
5. 自编消息框函数中添加icon图
6. 改进灰色棋子生成算法，在上局剩余棋子中生成

2013年11月26日：方块拼图1.3.1126

1. **添加生成灰色棋子时的动画**

2013年11月26日：方块拼图1.3b.1126

1. 解决两次以上改颜色无效的bug
2. 解决鼠标移到边缘没有预览棋子的bug
3. 解决了防止失败后不能保持旋转记录的bug

2013年11月27日：方块拼图1.4.1127

1. 更改算法使得每个账号可以拥有多个游戏记录
2. 更改算法使得名单链表全部使用动态空间
3. 解决了当前用户改昵称失败后显示当前玩家错误的bug
4. 解决了→无数个!!!由于解决bug和改进算法产生的bug

2013年11月28日：方块拼图1.5.1128

1. ***将重要变量改为全局变量***
2. 解决由于修改产生的大量bug
3. 更改算法，更改难度后将重新开局
4. **更改算法，可同时消除多个矩阵时有限消除面积大者**
5. **将系统选择框(MessageBox)改为自编选择框函数**

2013年11月29日：方块拼图1.6.1129

1. **添加商店功能，包括四种游戏道具**

2013年11月30日：方块拼图1.7.1130

1. 添加商店里新的两种游戏道具
2. **将多个游戏插图资源合并为一个文件以减少文件数**
3. 解决了L型棋子放置判断错误的bug

2013年12月1日：方块拼图1.8.1201

1. **将帮助功能改为动画介绍**

2013年12月3日：方块拼图1.9.1203

1. 将多种棋子/棋盘插图合并为一个图片文件
2. **鼠标经过不能放置的棋盘位置时棋子显示红叉**
3. ***将大部分全局变量删除***

2013年12月20日：方块拼图2.0.1220

1. **将EasyX输入框(InputBox)改为自编输入框函数，输入密码可以显示星号**

2013年12月21日：方块拼图2.1.1221

1. 游戏结束时显示分数

2013年12月24日：方块拼图2.2.1224

1. 最终修改

## 附录2 程序源代码

## 附录2.1 main.cpp

#include <graphics.h>

#include <conio.h>

#include <stdlib.h>

#include <time.h>

#include "Mmsystem.h"

#pragma comment(lib, "Winmm.lib")//添加音乐

#include "stdafx.h"

struct INFO

{

int level;

int score;

int pieceleft;

int dif;

int huiqi;

int RGBC;

int RGBC\_;

}nf;

#include "Click.h"

#include "Save.h"

#include "Block.h"

void load();

bool Game(char player[], bool read);

void ShowRank(char player[]);

void ShowHelp();

int ClickPlay(bool Has[], bool Grey[], INFO nf, int NewBlock, char player[], bool \*wheel);

void main()

{

IMAGE map;

loadimage(&map, "rc\\map.jpg");

char player[15]={0};

bool LoggedIn=0, play=1, read=0;

load(); //初始化

read=SaveReadClose(); //读取存档

while(!read && !LoggedIn)//如果没记录

{

if(ClickWelcome()) //欢迎界面

{

if(SaveLogIn(player))play=LoggedIn=1; //玩家信息

else putimage(0, 0, 640, 520, &map, 0, 500);

}

else play=0;

}

if(play)//玩游戏

{

solidrectangle(-1, 499, 640, 520);

while(Game(player, read));

}

cleardevice();

settextstyle(80, 0, "黑体"); settextcolor(WHITE);

outtextxy(220, 200, "再见！"); Sleep(1000);

closegraph();

}

bool Game(char player[], bool read)

{

int i;

FILE \*fp=fopen("rc\\temp.save" , "w"); fclose(fp); //首先清空临时文件

IMAGE back, map, playerback, difback, button, blocks, blockred;

loadimage(&blockred, "rc\\blockred.jpg");

loadimage(&blocks, "rc\\blocks.jpg");

loadimage(&map, "rc\\map.jpg");

loadimage(&button, "rc\\button.jpg", 80, 32);

bool Has[89]={0}, Grey[89]={0};

nf.RGBC=15539236, nf.RGBC\_=nf.RGBC;

nf.RGBC=BGR(nf.RGBC); nf.RGBC\_=BGR(nf.RGBC\_);

nf.level=1; nf.score=0; nf.pieceleft=15; nf.huiqi=0; nf.dif=1;

int NextBlock=rand()%7, NewBlock=NextBlock;

bool buy=0, usehuiqi=0, wheel=1; //是否购买道具

putimage(0, 0, 640, 500, &map, 0, 0);

getimage(&playerback, 168, 55, 150, 20);

getimage(&difback, 378, 55, 50, 20);

putimage(450, 7, &button); putimage(450, 40, &button);

putimage(540, 7, &button); putimage(540, 40, &button);

putimage(450, 222, &button); putimage(540, 222, &button);

settextcolor(WHITE); settextstyle(36, 0, "黑体");

outtextxy(50, 15, "可爱的方块拼图小游戏");

settextstyle(16, 0, "宋体"); outtextxy(470, 230, "悔棋 商店");

outtextxy(55, 58, "现在的玩家是："); outtextxy(170, 58, player);

outtextxy(335, 58, "难度：");

outtextxy(465, 15, "开局(R) 设置(S)"); outtextxy(465, 46, "帮助(H) 排名(L)");

settextcolor(BLACK); outtextxy(470, 110, "级别:");

outtextxy(470, 150, "金币(分数):"); outtextxy(470, 190, "需放棋子数:");

if(read)//读取存档

{

FILE \*fp=fopen("rc\\close.save", "r");

fscanf(fp, "%s");

fscanf(fp, "%d %d %d %d", &nf.level, &nf.score, &nf.pieceleft, &nf.dif);

for(i=11; i<89; i++)

{

fscanf(fp, "%d %d", &Has[i], &Grey[i]);

if(i%10==8){

int left=6+54\*(i%10-1), top=84+52\*(i/10-1);

if(Grey[i]) putimage(left-1, top-1, 49, 49, &blocks, 0, 50);

else if(Has[i] && !Grey[i]) putimage(left-1, top-1, &blockred);

else putimage(left-1, top-1, 49, 49, &blocks, 0, 0); }}

fclose(fp);

fp=fopen("rc\\close.save" , "w"); fclose(fp); //清空记录文件

}

while(1)//每进行一次游戏操作执行一次

{

int i, j;

BlockColor(&blockred, nf.RGBC, nf.RGBC\_);

putimage(575, 100, 57, 121, &map, 575, 100);

char slevel[3], spiece[3], sscore[8], sdif[5];

sprintf(slevel, "%d", nf.level); outtextxy(580, 110, slevel);

sprintf(sscore, "%d", nf.score); outtextxy(580, 150, sscore);

sprintf(spiece, "%d", nf.pieceleft); outtextxy(580, 190, spiece);

if(nf.dif==1)strcpy(sdif, "简单");

else if(nf.dif==2)strcpy(sdif, "中等");

else if(nf.dif==3)strcpy(sdif, "困难");

settextcolor(WHITE);

outtextxy(380, 58, sdif);

settextcolor(BLACK);

for(i=11; i<89; i++)//每次需要先根据Has和Grey把屏幕重新载入一遍

{

int left=6+54\*(i%10-1), top=84+52\*(i/10-1);

if(Grey[i])putimage(left-1, top-1, 49, 49, &blocks, 0, 50);

else if(Has[i] && !Grey[i])putimage(left-1, top-1, &blockred);

else putimage(left-1, top-1, 49, 49, &blocks, 0, 0);

if(i%10==8)i+=2; //防止出现19, 20等

}

if(!nf.pieceleft)//过关或第一次打开的情况

{

if(!nf.pieceleft){

time\_t start=time(NULL), finish=time(NULL);

mciSendString("play levelup from 0", NULL, 0, NULL);

ClickMSGSml("过关", "恭喜！您过关了！！！\n请等待音乐结束~", 2);

while(finish-start<5.5)finish=time(NULL);

nf.level++; nf.huiqi=0; }

nf.pieceleft=15+5\*(nf.level-1); //每一关剩余棋子数计算方法，可更改(注意：悔棋时用到本句话)

if(nf.level==11){

mciSendString("play win from 0", NULL, 0, NULL);

ClickMSGSml("通关", "大神，您通关了~~", 2);

SaveScore(nf.score, player); ShowRank(player); return 1; }

putimage(575, 100, 57, 121, &map, 575, 100); outtextxy(580, 150, sscore);

sprintf(slevel, "%d", nf.level); outtextxy(580, 110, slevel);

sprintf(spiece, "%d", nf.pieceleft); outtextxy(580, 190, spiece);

double grey; //grey灰棋子概率

if (nf.dif==3)grey=(49+2\*(nf.level-1))\*0.01; //灰色棋子算法，可更改

else if (nf.dif==2)grey=(34+2\*(nf.level-1))\*0.01;

else if (nf.dif==1)grey=(19+2\*(nf.level-1))\*0.01;

for(i=11; i<89; i++){

int left=6+54\*(i%10-1), top=84+52\*(i/10-1); j=5;

if(Has[i] && !Grey[i]){

while(j--){

rectangle(left, top, left+46, top+46); Sleep(10);

putimage(left-1, top-1, &blockred); Sleep(10); }

if(!(rand()%(int)(1/grey)))//上局遗留的部分转化为灰色

{

Has[i]=1; Grey[i]=1;

mciSendString("play stone from 0", NULL, 0, NULL);

putimage(left-1, top-1, 49, 49, &blocks, 0, 50); //放上灰色

}

else {Has[i]=0; putimage(left-1, top-1, 49, 49, &blocks, 0, 0); }}//上局遗留的没成灰色的

//剩余情况：Has&&Grey（原本就是灰色）和!Has（原本空格）都不需处理

if(i%10==8)i+=2; }

}//过关或第一次打开的情况结束

if(!buy){

NewBlock=NextBlock;

NextBlock=rand()%7; }

else buy=0;

if(!wheel){

FILE \*fp=fopen("rc\\temp.save", "a+");

fprintf(fp, "\n%s %d %d %d %d ", player, nf.level, nf.score, nf.pieceleft, NewBlock);

for(i=11; i<89; i++)fprintf(fp, "%d %d ", Has[i], Grey[i]);

fclose(fp); }

BlockDraw(Has, Grey, nf, NextBlock, 0, 0); //填写坐标为0表示要画出next

int CanPut=BlockCanPut(Has, Grey, NewBlock);

if(CanPut==0){

SaveScore(nf.score, player);

char str[50]; sprintf(str, "囧，您失败了……您的得分是%d。\n是否重玩？", nf.score);

if(ClickMSGSml("游戏结束", str, 3))

{ShowRank(player); return 1; }

else {ShowRank(player); return 0; }}

if(!strcmp(player, "computer")){

BlockDraw(Has, Grey, nf, NewBlock, 0, CanPut);

BlockDraw(Has, Grey, nf, NewBlock, 1, CanPut);

BlockDraw(Has, Grey, nf, NewBlock, 2, CanPut);

BlockDraw(Has, Grey, nf, NewBlock, 3, CanPut);

BlockDel(Has, Grey, &nf.score);

Sleep(200);

BlockDel(Has, Grey, &nf.score); nf.score+=(nf.dif\*2-1); nf.pieceleft--; nf.huiqi++; continue; }

reput:

switch(ClickPlay(Has, Grey, nf, NewBlock, player, &wheel))//对操作进行处理

{

case 0: if(ClickMSGSml("提示", "是否保存游戏记录？", 3))

{

FILE \*fp=fopen("rc\\close.save", "w"); //保存每次点击记录

fprintf(fp, "%s %d %d %d %d ", player, nf.level, nf.score, nf.pieceleft, nf.dif);

for(i=11; i<89; i++)fprintf(fp, "%d %d\n", Has[i], Grey[i]);

fclose(fp);

}

return 0;

case 1: BlockDel(Has, Grey, &nf.score); nf.score+=(nf.dif\*2-1); nf.pieceleft--; nf.huiqi++; break;

case 91:

if(ClickMSGSml("重玩", "确认重玩？", 3)){return 1; }

goto reput;

case 92://两行注释内容为查看棋盘上是否有棋子与是否有灰色棋子

// {char show[200]; sprintf(show, "%d %d %d %d %d %d %d %d\n%d %d %d %d %d %d %d %d\n%d %d %d %d %d %d %d %d\n%d %d %d %d %d %d %d %d\n%d %d %d %d %d %d %d %d\n%d %d %d %d %d %d %d %d\n%d %d %d %d %d %d %d %d\n%d %d %d %d %d %d %d %d", Has[11], Has[12], Has[13], Has[14], Has[15], Has[16], Has[17], Has[18], Has[21], Has[22], Has[23], Has[24], Has[25], Has[26], Has[27], Has[28], Has[31], Has[32], Has[33], Has[34], Has[35], Has[36], Has[37], Has[38], Has[41], Has[42], Has[43], Has[44], Has[45], Has[46], Has[47], Has[48], Has[51], Has[52], Has[53], Has[54], Has[55], Has[56], Has[57], Has[58], Has[61], Has[62], Has[63], Has[64], Has[65], Has[66], Has[67], Has[68], Has[71], Has[72], Has[73], Has[74], Has[75], Has[76], Has[77], Has[78], Has[81], Has[82], Has[83], Has[84], Has[85], Has[86], Has[87], Has[88]);

// MessageBox(0, show, "", MB\_OK); }

int Set; getimage(&back, 0, 0, 640, 520);

while(Set=ClickMSGSet(nf.dif))

{

if (Set==1)break;

else if(Set==101)SaveChange(1, player); //改昵称

else if(Set==102)SaveChange(2, player); //改密码，其实用不到player

else if(Set==103)SaveDelete(player); //删用户

else if(Set==104)SaveCreate(); //新用户

else if(Set==105){nf.dif=1; }

else if(Set==106){nf.dif=2; }

else if(Set==107){nf.dif=3; }

else if(Set==108){Set=ClickMSGRGB();

if(Set!=1){ nf.RGBC\_=nf.RGBC; nf.RGBC=Set;

BlockColor(&blockred, nf.RGBC, nf.RGBC\_); break; }}

if(Set==106||Set==107||Set==108){

for(i=0; i<89; i++) Has[i]=Grey[i]=0;

nf.level=1; nf.score=0; nf.pieceleft=15;

ClickMSGSml("提示", "游戏即将重新开局", 2); break; }

}

putimage(0, 0, &back);

putimage(168, 55, &playerback); settextcolor(WHITE); //以下两行针对Set==101

outtextxy(170, 58, player); settextcolor(BLACK);

if(Set==106||Set==107||Set==108){putimage(378, 55, &difback); break; }

for(i=11; i<89; i++){//本循环针对108

int left=6+54\*(i%10-1), top=84+52\*(i/10-1);

if(Has[i] && !Grey[i])putimage(left-1, top-1, &blockred);

if(i%10==8)i+=2; }

goto reput;

case 93: getimage(&back, 0, 0, 640, 520);

ShowHelp(); setfillcolor(RGB(45, 180, 160));

putimage(0, 0, &back); goto reput;

case 94: ShowRank(player); goto reput;

case 95: if(nf.huiqi){//本关开始时不能悔棋

if(!usehuiqi){ClickMSGSml("提示", "每次悔棋扣5分！", 2); usehuiqi=1; }

nf.huiqi--;

SaveBack(Has, Grey, &nf, &NewBlock, player);

putimage(575, 100, 57, 121, &map, 575, 100);

sprintf(slevel, "%d", nf.level); outtextxy(580, 110, slevel);

sprintf(sscore, "%d", nf.score); outtextxy(580, 150, sscore);

sprintf(spiece, "%d", nf.pieceleft); outtextxy(580, 190, spiece);

for(i=11; i<89; i++)

{

int left=6+54\*(i%10-1), top=84+52\*(i/10-1);

if(Grey[i])putimage(left-1, top-1, 49, 49, &blocks, 0, 50);

else if(Has[i] && !Grey[i])putimage(left-1, top-1, &blockred);

else putimage(left-1, top-1, 49, 49, &blocks, 0, 0);

if(i%10==8)i+=2;

}}

else ClickMSGSml("错误", "不能再悔棋了！", 1);

goto reput;

case 96:

getimage(&back, 0, 0, 640, 520);

int Tool;

while(Tool=ClickMSGTool())

{

if (Tool==1)break;

else if(Tool==111 && nf.score>=500){nf.score-=501; nf.huiqi--; for(i=11; i<89; i++)if(Grey[i])Grey[i]=Has[i]=0; break; }

else if(Tool==112 && nf.score>=200){nf.score-=201; nf.huiqi--; for(i=11; i<89; i++)Grey[i]=0; break; }

else if(Tool==113 && nf.score>=50){nf.score-=51; wheel=1; nf.huiqi--; break; }

else if(Tool==114 && nf.score>=150){nf.score-=51; NewBlock=7; break; }

else if(Tool==115 && nf.score>=150){nf.score-=51; NewBlock=8; break; }

else if(Tool==116 && nf.score>=150){nf.score-=101; NewBlock=9; break; }

else ClickMSGSml("错误", "金币不足", 1);

}

putimage(0, 0, &back); buy=1; nf.huiqi++; continue; //由于重新循环存档多一次，需增加一次悔棋机会

}

}

}

void ShowRank(char player[])

{

int i;

IMAGE back, message;

loadimage(&message, "rc\\message.jpg");

getimage(&back, 0, 0, 640, 520);

ClickMosaic();

putimage(100, 100, &message);

outtextxy(104, 104, "排名(分数相等按照时间排序)");

outtextxy(180, 130, "排名"); outtextxy(250, 130, "昵称" );

outtextxy(400, 130, "分数");

user \*H=NULL, \*p=NULL, \*file=NULL;

FILE \*fp=fopen("rc\\score.save", "r+");

H=(user\*)malloc(sizeof(user));

H->next=NULL;

p=H;

fscanf(fp, "\n");

while(!feof(fp))

{

file=(user\*)malloc(sizeof(user));

fscanf(fp, "%s\t%d", file->name, &file->score);

file->next=NULL;

p->next=file;

p=file;

}

p=H->next; //输出

char str[30];

for(i=160; i<(160+10\*20)&&p; i+=20, p=p->next)

{

sprintf(str, "%d", (i-140)/20);

outtextxy(190, i, str);

RECT r = {250, i, 440, i+20};

drawtext(p->name, &r, DT\_SINGLELINE);

sprintf(str, "%d", p->score);

outtextxy(400, i, str );

}

p=H->next; //查找自己

int rank=1;

for(; p&&strcmp(player, p->name); p=p->next)rank++;

if(!p)rank=0;

sprintf(str, "%s，您的排名是：%d", player, rank);

outtextxy(200, 370, str );

solidrectangle(-1, 499, 640, 520);

outtextxy(5, 502, "提示：点击右上角x关闭，或者在键盘按回车或Esc。");

ClickMSGRank();

putimage(0, 0, &back);

setfillcolor(RGB(45, 180, 160));

}

bool ShowHelpSleep(int time)

{

MOUSEMSG m;

IMAGE message;

loadimage(&message, "rc\\message.jpg");

int start=clock(), end=clock();

setfillcolor(RGB(45, 180, 160));

while(end-start<time){

if(kbhit()) {

mciSendString("play clickkb from 0", NULL, 0, NULL);

switch(getch()) {

case 13:case 27:setfillcolor(WHITE); return 1;

default:mciSendString("play error from 0", NULL, 0, NULL); break; }}

else if(MouseHit()) { m = GetMouseMsg();

if(m.x>528 && m.x<550 && m.y>100 && m.y<122){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击关闭对话框。"));

rectangle(528, 101, 548, 121);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); setfillcolor(WHITE); return 1; }}

else{

solidrectangle(-1, 499, 640, 520);

outtextxy(5, 502, \_T("提示：点击右上角×关闭，或者在键盘按回车或Esc。"));

putimage(527, 100, 23, 23, &message, 427, 0); }}

end=clock(); }

setfillcolor(WHITE);

return 0;

}

bool ShowHelpTextFlash(char help[])

{

solidrectangle(159, 134, 465, 154);

char helpchar[5]; int i=0;

while(i<(signed)strlen(help)){

sprintf(helpchar, "%c%c", help[i], help[i+1]);

outtextxy(160+8\*i, 135, helpchar); if(ShowHelpSleep(50))return 1; i+=2; }

if(ShowHelpSleep(500))return 1;

return 0;

}

void ShowHelp()

{

IMAGE help, message;

loadimage(&message, "rc\\message.jpg");

loadimage(&help, "rc\\help.jpg");

ClickMosaic();

putimage(100, 100, &message);

outtextxy(104, 104, "帮助"); putimage(101, 124, 448, 274, &help, 0, 0);

int i=4; setfillcolor(WHITE);

while(i--){

putimage(161, 174, 274, 70, &help, 0, 275); putimage(191, 254, 274, 70, &help, 0, 355);

if(ShowHelpSleep(500))return;

putimage(161, 174, 274, 70, &help, 280, 275); putimage(191, 254, 274, 70, &help, 280, 355);

if(ShowHelpSleep(500))return; }

solidrectangle(161, 174, 465, 324);

if(ShowHelpTextFlash("游戏开始时有一个棋盘。"))return;

putimage(200, 175, 217, 210, &help, 0, 439); if(ShowHelpSleep(1500))return;

if(ShowHelpTextFlash("同时一个浅色的棋子将会跟随您的鼠标。")) return;

putimage(200, 175, 217, 210, &help, 216, 439); if(ShowHelpSleep(1500))return;

if(ShowHelpTextFlash("您可也用键盘方向键控制位置。")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("鼠标点击左键或在键盘上按回车放置棋子。")) return;

putimage(200, 175, 217, 210, &help, 0, 647); if(ShowHelpSleep(1500))return;

if(ShowHelpTextFlash("根据游戏难度每放一个棋子得到不同加分。")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("简单加一分，中等加三分，困难加五分。")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("继续这样操作。")) return;

putimage(200, 175, 217, 210, &help, 216, 647); if(ShowHelpSleep(1500))return;

putimage(200, 175, 217, 210, &help, 0, 855); if(ShowHelpSleep(1500))return;

if(ShowHelpTextFlash("当３ｘ３面积内充满棋子时它们将被消除。")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("并且得到消除面积对应的额外金币。")) return;

putimage(200, 175, 217, 210, &help, 216, 855); if(ShowHelpSleep(1500))return;

putimage(200, 175, 217, 210, &help, 0, 1063); if(ShowHelpSleep(1500))return;

putimage(200, 175, 217, 210, &help, 216, 1063); if(ShowHelpSleep(1500))return;

solidrectangle(161, 174, 465, 384);

if(ShowHelpTextFlash("充满大于３ｘ３面积也会消除。")) return;

if(ShowHelpTextFlash("但如果同时以下情况：")) return;

putimage(200, 175, 217, 210, &help, 0, 1271); if(ShowHelpSleep(1500))return;

if(ShowHelpTextFlash("两块可消除区域有公共部分。")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("将消除面积更大的区域。")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("如果面积相等优先消除瘦高的。")) return;

putimage(200, 175, 217, 210, &help, 216, 1271); if(ShowHelpSleep(1500))return;

solidrectangle(161, 174, 465, 384);

if(ShowHelpTextFlash("鼠标右键或键盘空格可以旋转棋子。")) return;

putimage(200, 175, 217, 210, &help, 0, 1479); if(ShowHelpSleep(1500))return;

putimage(200, 175, 217, 210, &help, 216, 1479); if(ShowHelpSleep(1500))return;

putimage(200, 175, 217, 210, &help, 0, 1479); if(ShowHelpSleep(1500))return;

solidrectangle(161, 174, 465, 384);

if(ShowHelpTextFlash("当“需放棋子数”为零时过关。")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("此时未被消除的棋子按一定概率变灰。")) return; if(ShowHelpSleep(500))return;

putimage(200, 175, 217, 210, &help, 0, 1687); if(ShowHelpSleep(1500))return;

putimage(200, 175, 217, 210, &help, 216, 1687); if(ShowHelpSleep(1500))return;

if(ShowHelpTextFlash("变为灰色棋子后它们将不能被消除。")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("游戏越难变灰概率越大，概率也随关卡变大")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("如果棋盘上放不下更多的棋子则游戏结束。")) return; if(ShowHelpSleep(500))return;

if(ShowHelpTextFlash("游戏还有超多功能待您去探索（懒得写了）")) return; if(ShowHelpSleep(500))return;

solidrectangle(161, 174, 465, 384);

while(1){

putimage(161, 174, 274, 70, &help, 0, 275); putimage(191, 254, 274, 70, &help, 0, 355);

if(ShowHelpSleep(500))return;

putimage(161, 174, 274, 70, &help, 280, 275); putimage(191, 254, 274, 70, &help, 280, 355);

if(ShowHelpSleep(500))return; }

}

int ClickPlay(bool Has[], bool Grey[], INFO nf, int NewBlock, char player[], bool \*wheel)

{

MOUSEMSG m;

IMAGE backsml, blocks, blockred;

loadimage(&blockred, "rc\\blockred.jpg");

loadimage(&blocks, "rc\\blocks.jpg");

getimage(&backsml, 6, 84, 424, 410); //每次获取最大可能的范围

BlockColor(&blockred, nf.RGBC, nf.RGBC\_);

int co=44, codown, left, top, right, bottom, co\_=0, left\_, top\_, right\_, bottom\_;

left\_=left=168; top\_=top=240; right\_=right=214; bottom\_=bottom=286; //为键盘设计，初始格子是4排4列

int i, TypeBlock=0;

reput:

while(true)

{

if(kbhit())

{

mciSendString("play clickkb from 0", NULL, 0, NULL);

int get=getch();

if (get=='R' || get=='r')return 91;

else if(get=='S' || get=='s')return 92;

else if(get=='H' || get=='h')return 93;

else if(get=='L' || get=='l')return 94;

else if(get==13)

{

solidrectangle(left, top, right, bottom);

Sleep(100);

putimage(left-1, top-1, 49, 49, &blocks, 0, 0);

if(!BlockDraw(Has, Grey, nf, NewBlock, TypeBlock, co)){

putimage(6, 84, &backsml);

mciSendString("play error from 0", NULL, 0, NULL);

goto reput; }

getimage(&backsml, 6, 84, 430, 494);

return 1;

}

else if(get==27) return 0;

else if(get==' ')//旋转

{

TypeBlock++;

if(TypeBlock==4)TypeBlock=0;

putimage(6, 84, &backsml);

getimage(&backsml, 6, 84, 424, 410);

BlockTemp(Has, nf.RGBC, nf.RGBC\_, NewBlock, TypeBlock, co); //画出临时图

}

else if(get==0xE0)//小键盘上下左右

{

switch(getch())

{

case 72:if(top>84){co-=10; top-=52; bottom-=52; }break;

case 75:if(left>6){co-= 1; left-=54; right-=54; }break;

case 77:if(right<430){co+= 1; left+=54; right+=54; }break;

case 80:if(bottom<494){co+=10; top+=52; bottom+=52; }break;

}

if(co!=co\_&&co\_)putimage(6, 84, &backsml); //恢复上一格子边框颜色

BlockTemp(Has, nf.RGBC, nf.RGBC\_, NewBlock, TypeBlock, co); //画出临时图

co\_=co; left\_=left; top\_=top; right\_=right; bottom\_=bottom;

continue;

}

else mciSendString("play error from 0", NULL, 0, NULL); //break;

}

else if(MouseHit())

{

m = GetMouseMsg();

if(m.x>= 6 && m.x<=430&&m.y>= 84 && m.y<=494)

{ //6-52横向间距54, 每格46, 此处算上了4像素边框

if (m.x>= 3 && m.x<= 56){co = 1; left= 6; right= 52; }

else if(m.x>= 57 && m.x<=110){co = 2; left= 60; right=106; }//60-106

else if(m.x>=111 && m.x<=164){co = 3; left=114; right=160; }//114-160

else if(m.x>=165 && m.x<=218){co = 4; left=168; right=214; }//168-214

else if(m.x>=219 && m.x<=272){co = 5; left=222; right=268; }//222-268

else if(m.x>=273 && m.x<=326){co = 6; left=276; right=322; }//276-322

else if(m.x>=327 && m.x<=380){co = 7; left=330; right=376; }//330-376

else if(m.x>=381 && m.x<=434){co = 8; left=384; right=430; }//384-430

//84-136纵向间距52, 每格46, 此处算上了3像素边框

if (m.y>= 82 && m.y<=133){co+=10; top= 84; bottom=130; }

else if(m.y>=134 && m.y<=185){co+=20; top=136; bottom=182; }//136-182

else if(m.y>=186 && m.y<=237){co+=30; top=188; bottom=234; }//188-234

else if(m.y>=238 && m.y<=289){co+=40; top=240; bottom=286; }//240-286

else if(m.y>=290 && m.y<=341){co+=50; top=292; bottom=338; }//292-338

else if(m.y>=342 && m.y<=393){co+=60; top=344; bottom=390; }//344-390

else if(m.y>=394 && m.y<=445){co+=70; top=396; bottom=442; }//396-442

else if(m.y>=446 && m.y<=497){co+=80; top=448; bottom=494; }//448-494

if(m.uMsg==WM\_RBUTTONUP){//旋转

TypeBlock++;

if(TypeBlock==4)TypeBlock=0;

putimage(6, 84, &backsml);

getimage(&backsml, 6, 84, 424, 410); }

if(m.uMsg==WM\_MOUSEWHEEL && \*wheel){

putimage(6, 84, &backsml);

NewBlock=NewBlock>=6?0:(NewBlock+1); }

if(co!=co\_&&co\_)putimage(6, 84, &backsml); //鼠标不在同一个格子里时恢复上一格子边框颜色

if(m.uMsg==WM\_LBUTTONDOWN){

solidrectangle(left, top, right, bottom); //点下时框框变绿

codown=co; }

BlockTemp(Has, nf.RGBC, nf.RGBC\_, NewBlock, TypeBlock, co); //画出临时图

co\_=co; left\_=left; top\_=top; right\_=right; bottom\_=bottom;

if(m.uMsg==WM\_LBUTTONUP && codown==co)//鼠标点击和抬起在同一框内才能放置棋子

{

if(!BlockDraw(Has, Grey, nf, NewBlock, TypeBlock, co)){

putimage(6, 84, &backsml);

mciSendString("play error from 0", NULL, 0, NULL);

goto reput; } //没添加上重新来

mciSendString("play click from 0", NULL, 0, NULL);

getimage(&backsml, 6, 84, 430, 494);

\*wheel=0; //无论买没买任意棋子，到这里都失效

return 1;

}

}

else if(m.x>470 && m.x<540 && m.y>7 && m.y<39){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, "提示：点击重新开始游戏，或者在键盘按“R”。");

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 91; }}

else if(m.x>550 && m.x<620 && m.y>7 && m.y<39){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, "提示：点击设置游戏，或者在键盘按“S”。");

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 92; }}

else if(m.x>470 && m.x<540 && m.y>40 && m.y<72){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, "提示：点击显示游戏帮助，或者在键盘按“H”。");

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 93; }}

else if(m.x>550 && m.x<620 && m.y>40 && m.y<72){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, "提示：点击显示玩家排名，或者在键盘按“L”。");

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 94; }}

else if(m.x>450 && m.x<530 && m.y>222 && m.y<254){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, "提示：点击退回一步，悔棋一步扣除五分。");

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 95; }}

else if(m.x>540 && m.x<620 && m.y>222 && m.y<254){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, "提示：点击购买道具。");

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 96; }}

else {

for(i=11; i<89; i++)//取消棋子预览

{

left=6+54\*(i%10-1), top=84+52\*(i/10-1);

if(Grey[i])putimage(left-1, top-1, 49, 49, &blocks, 0, 50);

else if(Has[i] && !Grey[i])putimage(left-1, top-1, &blockred);

else putimage(left-1, top-1, 49, 49, &blocks, 0, 0);

if(i%10==8)i+=2; //防止出现19, 20等

}

solidrectangle(-1, 499, 640, 520); //移出区域后清除提示

outtextxy(5, 502, "提示：退出游戏点击右上角x，或者在键盘按Esc。"); }

}//鼠标点击结束

}//while结束

}

void load()

{

srand((unsigned)time(NULL));

mciSendString("open rc\\click.wav alias click" , NULL, 0, NULL);

mciSendString("open rc\\clickkb.wav alias clickkb" , NULL, 0, NULL);

mciSendString("open rc\\error.wav alias error" , NULL, 0, NULL);

mciSendString("open rc\\info.wav alias info" , NULL, 0, NULL);

mciSendString("open rc\\stone.wav alias stone" , NULL, 0, NULL);

mciSendString("open rc\\score.mp3 alias score" , NULL, 0, NULL);

mciSendString("open rc\\levelup.mp3 alias levelup" , NULL, 0, NULL);

mciSendString("open rc\\dead.mp3 alias dead" , NULL, 0, NULL);

mciSendString("open rc\\win.mp3 alias win" , NULL, 0, NULL);

initgraph(640, 520);

IMAGE map;

loadimage(&map, "rc\\map.jpg");

putimage(0, 0, 640, 520, &map, 0, 500); //初始界面

setbkmode(TRANSPARENT); setlinecolor(RGB(45, 180, 160));

setlinestyle(PS\_JOIN\_MITER|PS\_SOLID, 3); setfillcolor(RGB(45, 180, 160));

settextcolor(BLACK); settextstyle(16, 0, "宋体");

}

## 附录2.2 Save.h

#include "stdafx.h"

#include <graphics.h>

#include <conio.h>

#include <stdlib.h>

//定义结构体

struct user

{

char name[16];

char pass[16];

int score;

user \*next;

};

struct SAVETEMP

{

bool Has[89];

bool Grey[89];

int null; //没有用，但是没有他的话一旦悔棋level就丢了

int level;

int score;

int pieceleft;

char player[15];

int New;

SAVETEMP \*next;

};

bool SaveReadClose()

{

FILE \*fp=fopen("rc\\close.save", "r");

char fileplayer[15]={0};

bool read=0;

fscanf(fp, "%s", fileplayer);

if(fileplayer[0])//如果有记录

{

char str[50];

sprintf(str, "是否读取%s的游戏记录？", fileplayer);

if(ClickMSGSml("提示", str, 3))read=1;

else strcpy(fileplayer, "\0");

fclose(fp);

}

return read;

}

void SaveBack(bool Has[], bool Grey[], INFO \*nf, int \*NewBlock, char player[])//悔棋

{

int i;

SAVETEMP \*H=NULL, \*p=NULL, \*p\_=NULL, \*file;

FILE \*fp=fopen("rc\\temp.save", "r+");

H=(SAVETEMP\*)malloc(sizeof(SAVETEMP));

H->next=NULL;

p=H;

while(!feof(fp))

{

file=(SAVETEMP\*)malloc(sizeof(SAVETEMP));

fscanf(fp, "%s %d %d %d %d ", file->player, &(file->level), &(file->score), &(file->pieceleft), &(file->New));

for(int i=11; i<89; i++)fscanf(fp, "%d %d ", &file->Has[i], &file->Grey[i]);

file->next=NULL;

p->next=file;

p\_=p;

p=file;

}

p=p\_;

for(i=11; i<89; i++){Has[i]=p->Has[i]; Grey[i]=p->Grey[i]; }

(\*nf).level=p->level; (\*nf).score=p->score-5;

(\*nf).pieceleft=p->pieceleft; \*NewBlock=p->New;

strcpy(player, p->player);

fclose(fp);

p=H->next;

fp=fopen("rc\\temp.save", "w+");

for( ; p!=p\_->next; p=p->next)

{

fprintf(fp, "\n%s %d %d %d %d ", p->player, p->level, p->score-5, p->pieceleft, p->New);

for(i=11; i<89; i++)fprintf(fp, "%d %d ", p->Has[i], p->Grey[i]);

}

fclose(fp);

SAVETEMP \*del;

p=H;

while(p)

{

del=p;

p=p->next;

free(del);

}

}

void SaveRead(user \*H)//用于读取list.save文件，因为最常用，其他读取未制作函数

{

FILE \*fp=fopen("rc\\list.save", "r");

user \*p, \*file;

p=H;

fscanf(fp, "\n");

while(!feof(fp))

{

file=(user\*)malloc(sizeof(user));

fscanf(fp, "%s\t%s", file->name, file->pass);

file->next=NULL;

p->next=file;

p=file;

}

fclose(fp);

}

bool SaveInputPass(user \*p, user \*input)

{

if(p->pass[0]=='0')return 1; //没有密码

int time=6; //最多错5次

while(strcmp(p->pass, input->pass) && --time)

{

char str[50];

sprintf(str, "抱歉，输入错误，您还有%d次机会。", time);

if(time<5)ClickMSGSml("提示", str, 2);

if(!ClickInput("方块拼图", input->pass, "请输入【密码】。按取消重新输入昵称。", 1))return 0;

}

if(!time){ClickMSGSml("密码错误五次", "抱歉，请重新输入昵称。", 1); return 0; }

return 1; //只有输入正确才会返回1

}

void SaveNew(user \*input)

{

FILE \*fp=fopen("rc\\list.save", "a+");

if(!ClickInput("方块拼图", input->pass, "欢迎新用户，请设置【密码】。不设密码则输入0或不输入。", 1))

strcpy(input->pass, "0"); //点取消没有设置密码

if(input->pass[0]=='\0')strcpy(input->pass, "0"); //没设密码

fprintf(fp, "\n%s\t%s", input->name, input->pass); //存入信息

input->score=0;

fclose(fp);

}

void SaveNodeDel(user \*H)

{

user \*del, \*p=H;

while(p)

{

del=p; //del保存当前的p

p=p->next;

free(del);

}

}

bool SaveLogIn(char player[])

{

wrong://输入错误5次的起始点

user \*input=(user\*)malloc(sizeof(user));

if(!ClickInput("方块拼图", input->name, "请输入用户名。\n想看电脑自娱自乐请输入computer", 0))

return 0;

if(!input->name[0])

{

ClickMSGSml("错误", "昵称不能为空！", 1);

goto wrong;

}

user \*H=NULL, \*p=NULL;

H=(user\*)malloc(sizeof(user));

H->next=NULL;

SaveRead(H);

p=H->next;

for( ; p!=NULL; p=p->next)

{

if(!strcmp(p->name, input->name))//如果相同

{

if(!SaveInputPass(p, input))goto wrong; //1正常0回到输入昵称

FILE \*fp2=fopen("rc\\score.save", "r");

user \*file=(user\*)malloc(sizeof(user)); //读取老用户最高分当做输入值

fscanf(fp2, "\n");

do{

fscanf(fp2, "%s\t%d", file->name, &(file->score));

}while(strcmp(file->name, input->name)&&!feof(fp2));

if(strcmp(file->name, input->name))input->score=0; //没有成绩（建立账号后没有玩）

else input->score=file->score;

free(file); fclose(fp2);

break;

}

}

if(!p) SaveNew(input); //新用户

char str[50];

sprintf(str, "您好, %s。\n您的最高分是%d分!\n游戏即将开始!", input->name, input->score);

ClickMSGSml("您好", str, 2);

strcpy(player, input->name);

SaveNodeDel(H);

free(input);

return 1;

}

bool SaveChange(int type, char player[])//type=1改名字2改密码

{

wrong:

user \*input=(user\*)malloc(sizeof(user));

if(!ClickInput("方块拼图", input->name, "请输入用户名。", 0))

return 0;

if(!input->name[0] || !strcmp(input->name, "在这里输入…")){

ClickMSGSml("错误", "昵称不能为空！", 1); goto wrong; }

，实际上是SaveRead的变体

int find=0; char oldname[15];

user \*H=NULL, \*p, \*file, \*q;

FILE \*fp=fopen("rc\\list.save", "r");

H=(user\*)malloc(sizeof(user));

H->next=NULL;

p=H;

fscanf(fp, "\n");

while(!feof(fp))

{

file=(user\*)malloc(sizeof(user));

fscanf(fp, "%s\t%s", file->name, file->pass);

file->next=NULL;

p->next=file;

q=p;

p=file;

if(!strcmp(p->name, input->name))//如果相同

{

strcpy(oldname, p->name);

if(!SaveInputPass(p, input))goto wrong; //1正常0回到输入昵称

if(type==1)

{

find=1;

if(!ClickInput("方块拼图", input->name, "请输入新用户名。", 0))

return 0;

if(!input->name[0] || !strcmp(input->name, "在这里输入…")){

ClickMSGSml("错误", "昵称不能为空！", 1); goto wrong; }

strcpy(input->pass, p->pass);

input->next=NULL;

p=q;

p->next=input;

p=input;

}

else if(type==2)

{

find=1;

char secure[15]={"1"};

while(strcmp(secure, input->pass))

{

if(!ClickInput("方块拼图", input->pass, "请输入新【密码】，不设置直接确定。", 1))

return 0;

if(!strcmp(input->pass, "在这里输入…")||input->pass[0]=='\0')strcpy(input->pass, "0");

if(!ClickInput("方块拼图", secure, "请【确认】新【密码】。", 1))

return 0;

if(!strcmp(secure, "在这里输入…")||secure[0]=='\0' )strcpy(secure, "0");

if(strcmp(secure, input->pass)) ClickMSGSml("错误", "两次密码不一致，请重新输入！", 1);

}

strcpy(input->name, p->name);

input->next=NULL;

p=q;

p->next=input;

p=input;

}

}

}

fclose(fp);

if(!find){ClickMSGSml("错误", "该用户不存在，请更换！", 1); goto wrong; }

if(type==1)//改名时检查重名

{

user \*H\_=NULL;

H\_=(user\*)malloc(sizeof(user));

H\_->next=NULL;

SaveRead(H\_);

p=H\_;

for( ; p!=NULL; p=p->next)

{

if(!strcmp(p->name, input->name))//如果相同

{

fclose(fp);

strcpy(input->pass, "0");

ClickMSGSml("错误", "该名称已经存在，请【重新开始】！", 1);

goto wrong;

}

}

fclose(fp);

//如果是当前用户要把当前player改掉

if(!strcmp(player, oldname)) strcpy(player, input->name);

//不重名的话把score里面的名字全改了

fp=fopen("rc\\score.save", "r");

user \*H2=NULL;

H2=(user\*)malloc(sizeof(user));

H2->next=NULL;

p=H2;

fscanf(fp, "\n");

while(!feof(fp))

{

file=(user\*)malloc(sizeof(user));

fscanf(fp, "%s\t%d", file->name, &file->score);

file->next=NULL;

p->next=file;

p=file;

if(!strcmp(p->name, oldname))strcpy(p->name, input->name);

}

fclose(fp);

p=H2->next;

fp=fopen("rc\\score.save", "w+");

for( ; p!=NULL; p=p->next)

fprintf(fp, "\n%s\t%d", p->name, p->score);

fclose(fp);

SaveNodeDel(H2);

}

p=H->next;

fp=fopen("rc\\list.save", "w+");

for( ; p!=NULL; p=p->next)

fprintf(fp, "\n%s\t%s", p->name, p->pass);

fclose(fp);

SaveNodeDel(H); //free(input); input已经free过了

ClickMSGSml("成功", "修改成功！", 2);

return 1;

}

bool SaveCreate()

{

wrong:

user \*input=(user\*)malloc(sizeof(user));

if(!ClickInput("方块拼图", input->name, "请输入您要新建的用户名。", 0))

return 0;

if(!input->name[0] || !strcmp(input->name, "在这里输入…")){

ClickMSGSml("错误", "昵称不能为空！", 1); goto wrong; }

user \*H=NULL, \*p=NULL;

H=(user\*)malloc(sizeof(user));

H->next=NULL;

SaveRead(H);

p=H;

for( ; p!=NULL; p=p->next)

{

if(!strcmp(p->name, input->name))//如果相同

{

ClickMSGSml("错误", "该名称已经存在，请更换！", 1);

goto wrong;

}

}

SaveNew(input); //只有没找到才可能执行这步

char str[50];

sprintf(str, "您好, %s!\n您的最高分是%d分!", input->name, input->score);

ClickMSGSml("您好", str, 2);

SaveNodeDel(H); free(input);

return 1;

}

bool SaveDelete(char player[])

{

wrong:

user \*input=(user\*)malloc(sizeof(user));

if(!ClickInput("方块拼图", input->name, "请输入您要删除的用户名。", 0))

return 0;

if(!input->name[0]){

ClickMSGSml("错误", "昵称不能为空！", 1); goto wrong; }

user \*H=NULL, \*p=NULL;

H=(user\*)malloc(sizeof(user));

H->next=NULL;

SaveRead(H);

p=H;

for( ; p!=NULL; p=p->next)

{

if(p->next && !strcmp(p->next->name, input->name))//如果相同

{

if(!SaveInputPass(p->next, input))goto wrong; //1正常0回到输入昵称

if(!strcmp(player, input->name))ClickMSGSml("提示", "您正在删除当前用户，本账号将在关闭后删除~", 2);

else ClickMSGSml("提示", "该用户已经被删除！\n游戏记录将保存~", 2);

user \*q=p->next;

p->next=q->next;

free(q);

break;

}

}

if(!p){ClickMSGSml("错误", "该用户不存在，请更换！", 1); goto wrong; }

p=H->next;

FILE \*fp=fopen("rc\\list.save", "w+");

for( ; p!=NULL; p=p->next)

fprintf(fp, "\n%s\t%s", p->name, p->pass);

fclose(fp);

SaveNodeDel(H);

return 1;

}

void SaveScore(int score, char player[])

{

int i=0; bool add=0;

user \*H=NULL, \*p, \*q, \*file, \*input;

FILE \*fp=fopen("rc\\score.save", "r");

H=(user\*)malloc(sizeof(user));

H->next=NULL;

p=H;

fscanf(fp, "\n");

while(!feof(fp))

{

file=(user\*)malloc(sizeof(user));

fscanf(fp, "%s\t%d", file->name, &(file->score));

file->next=NULL;

p->next=file;

q=p;

p=file;

if(!add && score>p->score)//如果相同且为最高分

{

input=(user\*)malloc(sizeof(user));

strcpy(input->name, player);

strcpy(input->pass, p->pass);

input->score=score;

input->next=NULL;

q->next=input;

input->next=p;

add=1;

}

i++;

}

fclose(fp);

p=H->next;

fp=fopen("rc\\score.save", "w+");

for( ; p!=NULL; p=p->next)

fprintf(fp, "\n%s\t%d", p->name, p->score);

fclose(fp);

SaveNodeDel(H);

}

## 附录2.3 Click.h

#include "stdafx.h"

#include <graphics.h>

#include <conio.h>

#include <stdlib.h>

#include <time.h>

#include "Mmsystem.h"

#pragma comment(lib, "Winmm.lib")//添加音乐

void ClickMosaic()

{

int i, j;

IMAGE backmo;

getimage(&backmo, 0, 0, 640, 520);

IMAGE \*pimg=&backmo;

int width=pimg->getwidth(), height=pimg->getheight();

int redsum, greensum, bluesum, count, color;

// 获取指向显存的指针

DWORD\* pMem = GetImageBuffer(pimg);

// 处理每一个小方块

for (i=0; i<height; i+=15)

for (j=0; j<width; j+=15)

{

redsum = greensum = bluesum = count = 0;

// 求小方块的红、绿、蓝颜色值的和

for (int ty = min(i + 15, height) - 1; ty >= max(i, 0); ty--)

for (int tx = min(j +15, width) - 1; tx >= max(j, 0); tx--)

{

color = pMem[ty \* width + tx];

redsum += GetRValue(color);

greensum += GetGValue(color);

bluesum += GetBValue(color);

count++;

}

// 求红、绿、蓝颜色的平均值

redsum /= count;

greensum /= count;

bluesum /= count;

// 设置小方块内的每个像素为平均颜色值

color = RGB(redsum, greensum, bluesum);

for (ty = min(i + 15, height) - 1; ty >= max(i, 0); ty--)

for (int tx = min(j + 15, width) - 1; tx >= max(j, 0); tx--)

pMem[ty \* width + tx] = color;

}

putimage(0, 0, &backmo);

}

bool ClickInput(char title[], char input[], char help[], bool pass)

{

MOUSEMSG m;

IMAGE back2, icons, messagesml;

loadimage(&messagesml, "rc\\messagesml.jpg");

getimage(&back2, 0, 0, 640, 520);

ClickMosaic();

putimage(180, 180, &messagesml);

putimage(195, 220, 50, 50, &icons, 160, 0); outtextxy(270, 280, \_T("确定(Enter)")); outtextxy(370, 280, \_T("取消(Esc)"));

outtextxy(185, 185, title);

RECT r={190, 208, 445, 250};

drawtext(help, &r, DT\_WORDBREAK | DT\_WORD\_ELLIPSIS );

setfillcolor(WHITE);

fillrectangle(195, 245, 335, 271);

setfillcolor(RGB(45, 180, 160));

input[0]='\0';

int i=200, j=220;

while(true)

{

if(kbhit())

{

mciSendString("play clickkb from 0", NULL, 0, NULL);

char get[2]; get[0]=getch(); get[1]='\0';

if(get[0]==8 && strlen(input))input[strlen(input)-1]='\0';

else if(get[0]==13)return 1;

else if(get[0]==27){input[0]='\0'; return 0; }

else if(strlen(input)<15)strcat(input, get);

setfillcolor(WHITE);

fillrectangle(195, 245, 335, 271);

setfillcolor(RGB(45, 180, 160));

if(!pass)outtextxy(200, 250, input);

else

{

char passc[16];

for(int i=0; i<(signed)strlen(input); i++)passc[i]='\*';

passc[i]='\0';

outtextxy(200, 250, passc);

}

}

else if(MouseHit())

{

m = GetMouseMsg();

if(m.x>430 && m.x<452 && m.y>180 && m.y<202 || m.x>365 && m.x<450 && m.y>275 && m.y<300)

{

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击取消输入。"));

if(m.x>430 && m.x<452 && m.y>180 && m.y<202)rectangle(430, 180, 452, 202);

else rectangle(365, 275, 450, 300);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL);

putimage(0, 0, &back2); input[0]='\0'; return 0; }

}

else if(m.x>265 && m.x<360 && m.y>275 && m.y<300)

{

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击开始游戏。"));

rectangle(265, 275, 360, 300);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL);

putimage(0, 0, &back2); return 1; }

}

else

{

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击×或“确定”关闭，或者在键盘按回车或Esc。"));

setlinecolor(WHITE); //恢复上一个鼠标位置的边框

putimage(432, 180, 23, 23, &messagesml, 252, 0); rectangle(365, 275, 450, 300); rectangle(265, 275, 360, 300);

setlinecolor(RGB(45, 180, 160));

}

}

}

}

bool ClickWelcome()

{

MOUSEMSG m;

IMAGE icons;

loadimage(&icons, "rc\\icons.jpg");

while(true)

if(kbhit())

{

mciSendString("play clickkb from 0", NULL, 0, NULL);

switch(getch())

{

case 13:return 1; //getch()中13为回车

case 27:return 0;

default:mciSendString("play error from 0", NULL, 0, NULL); break;

}

}

else if(MouseHit())

{

m = GetMouseMsg();

if((m.x-175)\*(m.x-175)+(m.y-405)\*(m.y-405)<=2025)

{

if(m.uMsg==WM\_LBUTTONDOWN) putimage(119, 355, 109, 102, &icons, 0, 0);

else if(m.uMsg==WM\_LBUTTONUP)

{

mciSendString("play click from 0", NULL, 0, NULL);

loadimage(NULL, "rc\\start.jpg");

return 1;

}

}

else if(m.uMsg==WM\_LBUTTONUP)loadimage(NULL, "rc\\start.jpg");

}

}

bool ClickMSGSml(char title[], char str[], int type)

{

MOUSEMSG m;

IMAGE back2, icons, messagesml;

loadimage(&messagesml, "rc\\messagesml.jpg");

loadimage(&icons, "rc\\icons.jpg");

getimage(&back2, 0, 0, 640, 520);

ClickMosaic();

putimage(180, 180, &messagesml);

if(type==1) {

putimage(195, 220, 50, 50, &icons, 110, 0); outtextxy(410, 280, \_T("确定"));

mciSendString("play error from 0", NULL, 0, NULL); }

else if(type==2) {

putimage(195, 220, 50, 50, &icons, 110, 51); outtextxy(410, 280, \_T("确定"));

mciSendString("play info from 0", NULL, 0, NULL); }

else if(type==3){

putimage(195, 220, 50, 50, &icons, 160, 0); outtextxy(360, 280, \_T("是(Y)")); outtextxy(410, 280, \_T("否(N)"));

mciSendString("play info from 0", NULL, 0, NULL); }

RECT r={260, 220, 435, 334};

outtextxy(185, 185, title);

drawtext(str, &r, DT\_WORDBREAK | DT\_WORD\_ELLIPSIS );

while(true)

{

if(kbhit())

{

mciSendString("play clickkb from 0", NULL, 0, NULL);

switch(getch()) {

case 13:if(type==3){mciSendString("play error from 0", NULL, 0, NULL); break; }

case 27:putimage(0, 0, &back2); return 0;

case 'Y': case 'y': if(type==3){putimage(0, 0, &back2); return 1; }

case 'N': case 'n': if(type==3){putimage(0, 0, &back2); return 0; }

default:mciSendString("play error from 0", NULL, 0, NULL); break; }

}

else if(MouseHit())

{

m = GetMouseMsg();

if(m.x>430 && m.x<452 && m.y>180 && m.y<202 || m.x>405 && m.x<450 && m.y>275 && m.y<300)

{

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击关闭对话框。"));

if(m.x>430 && m.x<452 && m.y>180 && m.y<202)rectangle(430, 180, 452, 202);

else rectangle(405, 275, 450, 300);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL);

putimage(0, 0, &back2); return 0; }

}

else if(type==3 && m.x>355 && m.x<400 && m.y>275 && m.y<300)

{

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击继续程序。"));

rectangle(355, 275, 400, 300);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL);

putimage(0, 0, &back2); return 1; }

}

else

{

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击×或“确定”关闭，或者在键盘按回车或Esc。"));

setlinecolor(WHITE); //恢复上一个鼠标位置的边框

putimage(432, 180, 23, 23, &messagesml, 252, 0); rectangle(405, 275, 450, 300); rectangle(355, 275, 400, 300);

setlinecolor(RGB(45, 180, 160));

}

}

}

}

void ClickMSGRank()

{

MOUSEMSG m;

IMAGE message;

loadimage(&message, "rc\\message.jpg");

while(true)

{

if(kbhit())

{

mciSendString("play clickkb from 0", NULL, 0, NULL);

switch(getch())

{

case 13://getch()中13为回车

case 27:return ;

default:mciSendString("play error from 0", NULL, 0, NULL); break;

}

}

else if(MouseHit())

{

m = GetMouseMsg();

if(m.x>528 && m.x<550 && m.y>100 && m.y<122){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击关闭对话框。"));

rectangle(528, 101, 548, 121);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return ; }}

else

{

solidrectangle(-1, 499, 640, 520);

outtextxy(5, 502, \_T("提示：点击右上角×关闭，或者在键盘按回车或Esc。"));

putimage(527, 100, 23, 23, &message, 427, 0);

}

}

}

}

int ClickMSGSet(int dif)

{

MOUSEMSG m;

IMAGE message;

loadimage(&message, "rc\\message.jpg");

ClickMosaic();

putimage(100, 100, &message);

outtextxy(104, 104, \_T("设置"));

outtextxy(150, 150, \_T("修改昵称(C)")); outtextxy(150, 200, \_T("修改密码(P)"));

outtextxy(150, 250, \_T("删除账户(D)")); outtextxy(150, 300, \_T("新建账户(N)"));

setlinecolor(BLACK); setlinestyle(PS\_JOIN\_MITER|PS\_SOLID, 1);

line(270, 140, 270, 330);

setlinecolor(RGB(45, 180, 160)); setlinestyle(PS\_JOIN\_MITER|PS\_SOLID, 3);

outtextxy(300, 150, \_T("选择难度(选择后将重新开局)")); outtextxy(300, 180, \_T("简单(E)"));

outtextxy(380, 180, \_T("中等(M)")); outtextxy(460, 180, \_T("困难(H)"));

outtextxy(300, 230, \_T("改变棋子颜色"));

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击右上角×关闭，或者在键盘按回车或Esc。"));

setlinecolor(MAGENTA);

if(dif==1)rectangle(295, 175, 360, 200);

else if(dif==2)rectangle(375, 175, 440, 200);

else if(dif==3)rectangle(455, 175, 520, 200);

setlinecolor(RGB(45, 180, 160));

while(true)

{

if(kbhit())

{

mciSendString("play clickkb from 0", NULL, 0, NULL);

switch(getch())

{

case 13:case 27:return 1;

case 'C':case 'c':return 101;

case 'P':case 'p':return 102;

case 'D':case 'd':return 103;

case 'N':case 'n':return 104;

case 'E':case 'e':setlinecolor(WHITE); rectangle(375, 175, 440, 200); rectangle(455, 175, 520, 200); setlinecolor(RGB(45, 180, 160)); return 105;

case 'M':case 'm':setlinecolor(WHITE); rectangle(295, 175, 360, 200); rectangle(455, 175, 520, 200); setlinecolor(RGB(45, 180, 160)); return 106;

case 'H':case 'h':setlinecolor(WHITE); rectangle(295, 175, 360, 200); rectangle(375, 175, 440, 200); setlinecolor(RGB(45, 180, 160)); return 107;

default:mciSendString("play error from 0", NULL, 0, NULL); break;

}

}

else if(MouseHit())

{

m = GetMouseMsg();

if(m.x>528 && m.x<550 && m.y>100 && m.y<122){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击关闭对话框。"));

rectangle(528, 101, 548, 121);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 1; }}

else if(m.x>145 && m.x<250 && m.y>145 && m.y<170){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击修改玩家昵称，或者在键盘按“C”。"));

rectangle(145, 145, 250, 170);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 101; }}

else if(m.x>145 && m.x<250 && m.y>195 && m.y<220){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击修改玩家密码，或者在键盘按“P”。"));

rectangle(145, 195, 250, 220);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 102; }}

else if(m.x>145 && m.x<250 && m.y>245 && m.y<270){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击删除玩家账号，或者在键盘按“D”。"));

rectangle(145, 245, 250, 270);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 103; }}

else if(m.x>145 && m.x<250 && m.y>295 && m.y<320){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击新建玩家账号，或者在键盘按“N”。"));

rectangle(145, 295, 250, 320);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 104; }}

else if(m.x>295 && m.x<360 && m.y>175 && m.y<200 && dif!=1){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击设置游戏难度为简单，每放一个棋子加一分，或者在键盘按“E”。"));

rectangle(295, 175, 360, 200);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL);

setlinecolor(WHITE);

rectangle(375, 175, 440, 200); rectangle(455, 175, 520, 200);

setlinecolor(RGB(45, 180, 160)); return 105; }}

else if(m.x>375 && m.x<440 && m.y>175 && m.y<200 && dif!=2){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击设置游戏难度为中等，每放一个棋子加三分，或者在键盘按“M”。"));

rectangle(375, 175, 440, 200);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL);

setlinecolor(WHITE);

rectangle(295, 175, 360, 200); rectangle(455, 175, 520, 200);

setlinecolor(RGB(45, 180, 160)); return 106; }}

else if(m.x>455 && m.x<520 && m.y>175 && m.y<200 && dif!=3){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击设置游戏难度为困难，每放一个棋子加五分，或者在键盘按“H”。"));

rectangle(455, 175, 520, 200);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL);

setlinecolor(WHITE);

rectangle(295, 175, 360, 200); rectangle(375, 175, 440, 200);

setlinecolor(RGB(45, 180, 160)); return 107; }}

else if(m.x>295 && m.x<400 && m.y>225 && m.y<250){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击设置棋子颜色。"));

rectangle(295, 225, 400, 250);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 108; }}

else

{

solidrectangle(-1, 499, 640, 520);

outtextxy(5, 502, \_T("提示：点击右上角×关闭，或者在键盘按回车或Esc。"));

setlinecolor(WHITE); //恢复上一个鼠标位置的边框

putimage(527, 100, 23, 23, &message, 427, 0); rectangle(295, 225, 400, 250);

rectangle(145, 145, 250, 170); rectangle(145, 195, 250, 220);

rectangle(145, 245, 250, 270); rectangle(145, 295, 250, 320);

if(dif!=1)rectangle(295, 175, 360, 200);

if(dif!=2)rectangle(375, 175, 440, 200);

if(dif!=3)rectangle(455, 175, 520, 200);

setlinecolor(RGB(45, 180, 160));

}

}

}

}

int ClickToolFlash(int type, int time)

{

MOUSEMSG m;

int start=clock(), end=clock();

while(end-start<time){

if(MouseHit()){ m= GetMouseMsg();

if(m.x<145 || m.x>250 || m.y<115+type\*30 || m.y>140+type\*30)return 0;

if(m.uMsg==WM\_LBUTTONUP)return 1; }

end=clock(); }

return -1;

}

int ClickMSGTool()

{

MOUSEMSG m;

IMAGE tool, toolseven, message;

loadimage(&message, "rc\\message.jpg");

loadimage(&tool, "rc\\tool.jpg", 340, 850);

loadimage(&toolseven, "rc\\toolseven.jpg", 200, 170);

ClickMosaic();

putimage(100, 100, &message);

outtextxy(104, 104, \_T("商店"));

outtextxy(150, 150, \_T("灰棋光光(G)")); outtextxy(150, 180, \_T("速速解冻(R)"));

outtextxy(150, 210, \_T("我选棋子(A)")); outtextxy(150, 240, \_T("只来一块(O)"));

outtextxy(150, 270, \_T("可爱橡皮(E)")); outtextxy(150, 300, \_T("强力橡皮(S)"));

setlinecolor(BLACK); setlinestyle(PS\_JOIN\_MITER|PS\_SOLID, 1);

line(270, 140, 270, 330);

setlinecolor(RGB(45, 180, 160)); setlinestyle(PS\_JOIN\_MITER|PS\_SOLID, 3);

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击右上角x关闭，或者在键盘按回车或Esc。"));

while(true)

{

if(kbhit())

{

mciSendString("play clickkb from 0", NULL, 0, NULL);

switch(getch())

{

case 13:case 27:return 1;

case 'G':case 'g':return 111;

case 'R':case 'r':return 112;

case 'A':case 'a':return 113;

case 'O':case 'o':return 114;

case 'E':case 'e':return 115;

case 'S':case 's':return 116;

default:mciSendString("play error from 0", NULL, 0, NULL); break;

}

}

else if(MouseHit())

{

m = GetMouseMsg(); int Tool;

if(m.x>528 && m.x<550 && m.y>100 && m.y<122){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击关闭对话框。"));

rectangle(528, 101, 548, 121);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 1; }}

else if(m.x>145 && m.x<250 && m.y>145 && m.y<170){

setfillcolor(WHITE);

solidrectangle(300, 140, 530, 390); setfillcolor(RGB(45, 180, 160));

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击购买消除全部灰色棋子的道具，或者在键盘按“G”。"));

rectangle(145, 145, 250, 170);

char help[]={"购买此道具后屏幕上所有灰色棋子将被消除，购买本工具将会消耗500个金币。"};

RECT r = {300, 140, 530, 350}; drawtext(help, &r, DT\_WORDBREAK | DT\_WORD\_ELLIPSIS );

while(1)

{

putimage(315, 220, 170, 170, &tool, 0, 0); Tool=ClickToolFlash(1, 1000);

if(Tool==1)return 111; else if(Tool==0)break;

putimage(315, 220, 170, 170, &tool, 170, 0); settextcolor(WHITE);

outtextxy(372, 320, "灰棋光光！"); settextcolor(BLACK);

Tool=ClickToolFlash(1, 1000);

if(Tool==1)return 111; else if(Tool==0)break;

putimage(315, 220, 170, 170, &tool, 170, 170); Tool=ClickToolFlash(1, 2000);

if(Tool==1)return 111; else if(Tool==0)break;

}

if(m.uMsg==WM\_LBUTTONUP){mciSendString("play click from 0", NULL, 0, NULL); return 111; }}

else if(m.x>145 && m.x<250 && m.y>175 && m.y<200){

setfillcolor(WHITE);

solidrectangle(300, 140, 530, 390); setfillcolor(RGB(45, 180, 160));

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击购买解除灰色棋子的道具，或者在键盘按“R”。"));

rectangle(145, 175, 250, 200);

char help[]={"购买此道具后屏幕上所有灰色棋子将可以被消除，购买本工具将会消耗200个金币。"};

RECT r = {300, 140, 530, 350}; drawtext(help, &r, DT\_WORDBREAK | DT\_WORD\_ELLIPSIS );

while(1)

{

putimage(315, 220, 170, 170, &tool, 0, 0); Tool=ClickToolFlash(2, 1000);

if(Tool==1)return 112; else if(Tool==0)break;

putimage(315, 220, 170, 170, &tool, 170, 0); settextcolor(WHITE);

outtextxy(372, 320, "速速解冻！"); settextcolor(BLACK);

Tool=ClickToolFlash(2, 1000);

if(Tool==1)return 112; else if(Tool==0)break;

putimage(315, 220, 170, 170, &tool, 0, 170); Tool=ClickToolFlash(2, 2000);

if(Tool==1)return 112; else if(Tool==0)break;

}

if(m.uMsg==WM\_LBUTTONUP){mciSendString("play click from 0", NULL, 0, NULL); return 112; }}

else if(m.x>145 && m.x<250 && m.y>205 && m.y<230){

setfillcolor(WHITE);

solidrectangle(300, 140, 530, 390); setfillcolor(RGB(45, 180, 160));

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击购买任选棋子样式的道具，或者在键盘按“A”。"));

rectangle(145, 205, 250, 230);

char help[]={"购买此道具后您可以通过鼠标滚轮任意选择一次当前棋子的样式，购买本工具将会消耗50个金币。悔棋后不能恢复道具。"};

RECT r = {300, 140, 530, 350}; drawtext(help, &r, DT\_WORDBREAK | DT\_WORD\_ELLIPSIS );

putimage(315, 210, &toolseven);

if(m.uMsg==WM\_LBUTTONUP){mciSendString("play click from 0", NULL, 0, NULL); return 113; }}

else if(m.x>145 && m.x<250 && m.y>235 && m.y<260){

setfillcolor(WHITE);

solidrectangle(300, 140, 530, 390); setfillcolor(RGB(45, 180, 160));

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击购买只占一格的棋子，或者在键盘按“O”。"));

rectangle(145, 235, 250, 260);

char help[]={"购买此道具后当前棋子将变为单位棋子一次，该棋子可以放在任意空位置，购买本工具将会扣除50分。"};

RECT r = {300, 140, 530, 350}; drawtext(help, &r, DT\_WORDBREAK | DT\_WORD\_ELLIPSIS );

while(1)

{

putimage(315, 220, 170, 170, &tool, 0, 0); Tool=ClickToolFlash(4, 1000);

if(Tool==1)return 114; else if(Tool==0)break;

putimage(315, 220, 170, 170, &tool, 0, 340); Tool=ClickToolFlash(4, 1000);

if(Tool==1)return 114; else if(Tool==0)break;

putimage(315, 220, 170, 170, &tool, 170, 340); Tool=ClickToolFlash(4, 2000);

if(Tool==1)return 114; else if(Tool==0)break;

}

if(m.uMsg==WM\_LBUTTONUP){mciSendString("play click from 0", NULL, 0, NULL); return 114; }}

else if(m.x>145 && m.x<250 && m.y>265 && m.y<290){

setfillcolor(WHITE);

solidrectangle(300, 140, 530, 390); setfillcolor(RGB(45, 180, 160));

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击购买一块可擦除单位棋子的橡皮，或者在键盘按“O”。"));

rectangle(145, 265, 250, 290);

char help[]={"购买此道具后您可以擦除不是灰色的单位棋子一次，购买本工具将会扣除50分。"};

RECT r = {300, 140, 530, 350}; drawtext(help, &r, DT\_WORDBREAK | DT\_WORD\_ELLIPSIS );

while(1)

{

putimage(315, 220, 170, 170, &tool, 0, 0); Tool=ClickToolFlash(5, 1000);

if(Tool==1)return 115; else if(Tool==0)break;

putimage(315, 218, 170, 170, &tool, 0, 510); Tool=ClickToolFlash(5, 1000);

if(Tool==1)return 115; else if(Tool==0)break;

putimage(315, 218, 170, 170, &tool, 170, 510); Tool=ClickToolFlash(5, 2000);

if(Tool==1)return 115; else if(Tool==0)break;

}

if(m.uMsg==WM\_LBUTTONUP){mciSendString("play click from 0", NULL, 0, NULL); return 115; }}

else if(m.x>145 && m.x<250 && m.y>295 && m.y<320){

setfillcolor(WHITE);

solidrectangle(300, 140, 530, 390); setfillcolor(RGB(45, 180, 160));

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击购买一块可擦除单位棋子的橡皮，或者在键盘按“O”。"));

rectangle(145, 295, 250, 320);

char help[]={"购买此道具后您可以擦除任意3x3面积内的棋子一次，购买本工具将会扣除100分。"};

RECT r = {300, 140, 530, 350}; drawtext(help, &r, DT\_WORDBREAK | DT\_WORD\_ELLIPSIS );

while(1)

{

putimage(315, 220, 170, 170, &tool, 0, 0); Tool=ClickToolFlash(6, 1000);

if(Tool==1)return 116; else if(Tool==0)break;

putimage(315, 218, 170, 170, &tool, 0, 680); Tool=ClickToolFlash(6, 1000);

if(Tool==1)return 116; else if(Tool==0)break;

putimage(315, 218, 170, 170, &tool, 170, 680); Tool=ClickToolFlash(6, 2000);

if(Tool==1)return 116; else if(Tool==0)break;

}

if(m.uMsg==WM\_LBUTTONUP){mciSendString("play click from 0", NULL, 0, NULL); return 116; }}

else

{

solidrectangle(-1, 499, 640, 520);

outtextxy(5, 502, \_T("提示：点击右上角×关闭，或者在键盘按回车或Esc。"));

setlinecolor(WHITE); //恢复上一个鼠标位置的边框

putimage(527, 100, 23, 23, &message, 427, 0); setfillcolor(WHITE);

solidrectangle(300, 140, 530, 390); setfillcolor(RGB(45, 180, 160));

rectangle(145, 145, 250, 170); rectangle(145, 175, 250, 200);

rectangle(145, 205, 250, 230); rectangle(145, 235, 250, 260);

rectangle(145, 265, 250, 290); rectangle(145, 295, 250, 320);

setlinecolor(RGB(45, 180, 160));

}

}

}

}

int ClickMSGRGB()

{

MOUSEMSG m;

IMAGE colorimg, message;

loadimage(&message, "rc\\message.jpg");

loadimage(&colorimg, "rc\\color.jpg");

putimage(100, 100, &message);

putimage(140, 140, &colorimg);

outtextxy(104, 104, \_T("设置颜色"));

int color;

while(true)

{

if(kbhit())

{

mciSendString("play clickkb from 0", NULL, 0, NULL);

switch(getch())

{

case 13:

case 27:return 1;

default:mciSendString("play error from 0", NULL, 0, NULL); break;

}

}

else if(MouseHit())

{

m = GetMouseMsg();

if(m.x>528 && m.x<550 && m.y>100 && m.y<122){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击关闭对话框。"));

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL); return 1; }}

else if(m.x>=140 && m.x<=380 && m.y>=140 && m.y<=380){

solidrectangle(-1, 499, 640, 520); outtextxy(5, 502, \_T("提示：点击选择颜色(看我的颜色多炫~~)。"));

color=HSLtoRGB((float)((m.x-140)\*1.5), (float)(380-m.y)/240, 0.5);

setlinecolor(color); setfillcolor(color);

solidrectangle(400, 140, 450, 190);

if(m.uMsg==WM\_LBUTTONUP){

mciSendString("play click from 0", NULL, 0, NULL);

setlinecolor(RGB(45, 180, 160)); setfillcolor(RGB(45, 180, 160));

return color; }}

else

{

setlinecolor(RGB(45, 180, 160)); setfillcolor(RGB(45, 180, 160)); solidrectangle(-1, 499, 640, 520);

outtextxy(5, 502, \_T("提示：点击右上角×关闭，或者在键盘按回车或Esc。"));

}

}

}

}

## 附录2.4 Block.h

#include "stdafx.h"

#include <graphics.h>

#include <conio.h>

#include <stdlib.h>

#include <time.h>

void BlockColor(IMAGE \*pimg, int RGBC, int RGBC\_)

{

int width = pimg->getwidth(); // 图像的宽

int height = pimg->getheight(); // 图像的高

int i, j;

DWORD\* pMem = GetImageBuffer(pimg);

float H, S, L; RGBtoHSL(RGBC\_, &H, &S, &L);

L=L+(float)0.3;

for (i=0; i<height; i++)

for (j=0; j<width; j++)//太坑爹了！pMem[i\*width+j]中RGB存放顺序是BGR

{

if(pMem[i\*width+j]==15539236||pMem[i\*width+j]==BGR(RGBC\_)||pMem[i\*width+j]==BGR(HSLtoRGB(H, S, L)))

pMem[i\*width+j]=BGR(RGBC);

}

}

void BlockTemp(bool Has[], int RGBC, int RGBC\_, int NewBlock, int TypeBlock, int co)

{

IMAGE blocks, blockredtemp;

loadimage(&blockredtemp, "rc\\blockred.jpg");

loadimage(&blocks, "rc\\blocks.jpg");

int left=6+54\*(co%10-1), top=84+52\*(co/10-1);

float H, S, L; RGBtoHSL(RGBC, &H, &S, &L);

L=L+(float)0.3;

BlockColor(&blockredtemp, HSLtoRGB(H, S, L), RGBC\_);

if(NewBlock==0 && (TypeBlock==0 || TypeBlock==2)) {

if(co%10<=5 && Has[co]==0 && Has[co+1]==0 && Has[co+2]==0 && Has[co+3]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1+108, top-1, &blockredtemp);

putimage(left-1+162, top-1, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=6)putimage(left-1+108, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=5)putimage(left-1+162, top-1, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==0 && (TypeBlock==1 || TypeBlock==3)) {

if(co/10<=5 && Has[co]==0 && Has[co+10]==0 && Has[co+20]==0 && Has[co+30]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp);

putimage(left-1, top-1+104, &blockredtemp);

putimage(left-1, top-1+156, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50);

if(co/10<=6)putimage(left-1, top-1+104, 49, 49, &blocks, 50, 50);

if(co/10<=5)putimage(left-1, top-1+156, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==1 && TypeBlock==0) {

if(co%10<=7 && co/10>=3 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co-20]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp);

putimage(left-1, top-1-104, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50);

if(co/10>=3)putimage(left-1, top-1-104, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==1 && TypeBlock==1) {

if(co%10<=6 && co/10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co+2]==0 && Has[co+10]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp);

putimage(left-1+108, top-1, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50);

if(co%10<=6)putimage(left-1+108, top-1, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==1 && TypeBlock==2) {

if(co%10>=2 && co/10<=6 && Has[co]==0 && Has[co-1]==0 && Has[co+10]==0 && Has[co+20]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1-54, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp);

putimage(left-1, top-1+104, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10>=2)putimage(left-1-54, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50);

if(co/10<=6)putimage(left-1, top-1+104, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==1 && TypeBlock==3) {

if(co%10>=3 && co/10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co-2]==0 && Has[co-10]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1-54, top-1, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp);

putimage(left-1-108, top-1, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10>=2)putimage(left-1-54, top-1, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50);

if(co%10>=3)putimage(left-1-108, top-1, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==2 && TypeBlock==0) {

if(co%10>=2 && co/10>=3 && Has[co]==0 && Has[co-1]==0 && Has[co-10]==0 && Has[co-20]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1-54, top-1, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp);

putimage(left-1, top-1-104, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10>=2)putimage(left-1-54, top-1, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50);

if(co/10>=3)putimage(left-1, top-1-104, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==2 && TypeBlock==1) {

if(co%10<=6 && co/10>=2 && Has[co]==0 && Has[co-10]==0 && Has[co+1]==0 && Has[co+2]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp);

putimage(left-1+108, top-1, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50);

if(co%10<=6)putimage(left-1+108, top-1, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==2 && TypeBlock==2) {

if(co%10<=7 && co/10<=6 && Has[co]==0 && Has[co+1]==0 && Has[co+10]==0 && Has[co+20]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp);

putimage(left-1, top-1+104, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50);

if(co/10<=6)putimage(left-1, top-1+104, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==2 && TypeBlock==3) {

if(co%10>=3 && co/10<=7 && Has[co]==0 && Has[co-1]==0 && Has[co-2]==0 && Has[co+10]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1-54, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp);

putimage(left-1-108, top-1, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10>=2)putimage(left-1-54, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50);

if(co%10>=3)putimage(left-1-108, top-1, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==3 && TypeBlock==0) {

if(co/10>=2 && co%10<=7 && co%10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co+1]==0 && Has[co-10]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1-54, top-1, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co%10>=2)putimage(left-1-54, top-1, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==3 && TypeBlock==1) {

if(co/10>=2 && co/10<=7 && co%10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co+10]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==3 && TypeBlock==2) {

if(co/10<=7 && co%10<=7 && co%10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co+1]==0 && Has[co+10]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1-54, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co%10>=2)putimage(left-1-54, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==3 && TypeBlock==3) {

if(co/10>=2 && co/10<=7 && co%10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co+10]==0 && Has[co-10]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp);

putimage(left-1-54, top-1, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50);

if(co%10>=2)putimage(left-1-54, top-1, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==4 && (TypeBlock==0 || TypeBlock==2)) {

if(co%10<=7 && co%10>=2 && co/10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co-10]==0 && Has[co-9]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1-54, top-1, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp);

putimage(left-1+54, top-1-52, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10>=2)putimage(left-1-54, top-1, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50);

if(co%10<=7 && co/10>=2)putimage(left-1+54, top-1-52, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==4 && (TypeBlock==1 || TypeBlock==3)) {

if(co%10<=7 && co/10>=2 && co/10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co+11]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp);

putimage(left-1+54, top-1+52, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50);

if(co%10<=7 && co/10<=7)putimage(left-1+54, top-1+52, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==5 && (TypeBlock==0 || TypeBlock==2)) {

if(co%10<=7 && co%10>=2 && co/10>=2 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co-11]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1, top-1-52, &blockredtemp);

putimage(left-1-54, top-1-52, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co/10>=2)putimage(left-1, top-1-52, 49, 49, &blocks, 50, 50);

if(co%10>=2 && co/10>=2)putimage(left-1-54, top-1-52, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==5 && (TypeBlock==1 || TypeBlock==3)) {

if(co%10<=7 && co/10>=2 && co/10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co+10]==0 && Has[co-9]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp);

putimage(left-1+54, top-1-52, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50);

if(co%10<=7 && co/10>=2)putimage(left-1+54, top-1-52, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==6) {

if(co/10<=7 && co%10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co+10]==0 && Has[co+11]==0){

putimage(left-1, top-1, &blockredtemp);

putimage(left-1+54, top-1, &blockredtemp);

putimage(left-1, top-1+52, &blockredtemp);

putimage(left-1+54, top-1+52, &blockredtemp); }

else{

putimage(left-1, top-1, 49, 49, &blocks, 50, 50);

if(co%10<=7)putimage(left-1+54, top-1, 49, 49, &blocks, 50, 50);

if(co/10<=7)putimage(left-1, top-1+52, 49, 49, &blocks, 50, 50);

if(co%10<=7 && co/10<=7)putimage(left-1+54, top-1+52, 49, 49, &blocks, 50, 50); } }

else if(NewBlock==7) putimage(left-1, top-1, &blockredtemp);

else if(NewBlock==8) putimage(left-1, top-1, 49, 49, &blocks, 0, 0);

else if(NewBlock==9)//消除3x3棋子

{

putimage(left-1, top-1, 49, 49, &blocks, 0, 0);

if(co/10>=2 && co%10>=2)putimage(left-1-54, top-1-52, 49, 49, &blocks, 0, 0);

if(co/10>=2) putimage(left-1, top-1-52, 49, 49, &blocks, 0, 0);

if(co/10>=2 && co%10<=7)putimage(left-1+54, top-1-52, 49, 49, &blocks, 0, 0);

if(co%10>=2) putimage(left-1-54, top-1, 49, 49, &blocks, 0, 0);

if(co%10<=7) putimage(left-1+54, top-1, 49, 49, &blocks, 0, 0);

if(co/10<=7 && co%10>=2)putimage(left-1-54, top-1+52, 49, 49, &blocks, 0, 0);

if(co/10<=7) putimage(left-1, top-1+52, 49, 49, &blocks, 0, 0);

if(co/10<=7 && co%10<=7)putimage(left-1+54, top-1+52, 49, 49, &blocks, 0, 0);

}

fillrectangle(-1, 499, 640, 520); //输出提示

char str[100];

sprintf(str, "提示：目前位于第%d行第%d列，鼠标左键/回车确认，右键/空格旋转，方向键移动。", co/10, co%10);

settextstyle(16, 0, \_T("宋体"));

outtextxy(5, 502, str);

}

bool BlockDraw(bool Has[], bool Grey[], INFO nf, int NewNextB, int TypeBlock, int co)//返回0不能放置，1成功

{//加入棋子

IMAGE map, blocks, blockred;

loadimage(&blockred, "rc\\blockred.jpg");

loadimage(&blocks, "rc\\blocks.jpg");

loadimage(&map, "rc\\map.jpg");

BlockColor(&blockred, nf.RGBC, nf.RGBC\_);

if(!co){

putimage(432, 288, 208, 212, &map, 432, 288);

outtextxy(470, 300, \_T("下个棋子:")); }

int left=6+54\*(co%10-1), top=84+52\*(co/10-1);

if(NewNextB==0 && (!co || (TypeBlock==0 || TypeBlock==2) && co%10<=5 && Has[co]==0 && Has[co+1]==0 && Has[co+2]==0 && Has[co+3]==0))

{

if(!co){left=430; top=374; putimage(left-1, top-1, 49, 49, &blocks, 50, 0); }

else putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1+108, top-1, &blockred); putimage(left-1+162, top-1, &blockred);

if(co){Has[co]=Has[co+1]=Has[co+2]=Has[co+3]=1; }

}

else if(NewNextB==0 && (TypeBlock==1 || TypeBlock==3) && co/10<=5 && Has[co]==0 && Has[co+10]==0 && Has[co+20]==0 && Has[co+30]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1, top-1+52, &blockred);

putimage(left-1, top-1+104, &blockred); putimage(left-1, top-1+156, &blockred);

Has[co]=Has[co+10]=Has[co+20]=Has[co+30]=1;

}

//第2种 L型

else if(NewNextB==1 && (!co || TypeBlock==0 && co%10<=7 && co/10>=3 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co-20]==0))

{

if(!co){left=480; top=425; putimage(left-1, top-1, 49, 49, &blocks, 50, 0); }

else putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1, top-1-52, &blockred); putimage(left-1, top-1-104, &blockred);

if(co){Has[co]=Has[co+1]=Has[co-10]=Has[co-20]=1; }

}

else if(NewNextB==1 && TypeBlock==1 && co%10<=6 && co/10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co+2]==0 && Has[co+10]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1, top-1+52, &blockred); putimage(left-1+108, top-1, &blockred);

Has[co]=Has[co+1]=Has[co+2]=Has[co+10]=1;

}

else if(NewNextB==1 && TypeBlock==2 && co%10>=2 && co/10<=6 && Has[co]==0 && Has[co-1]==0 && Has[co+10]==0 && Has[co+20]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1-54, top-1, &blockred);

putimage(left-1, top-1+52, &blockred); putimage(left-1, top-1+104, &blockred);

Has[co]=Has[co-1]=Has[co+10]=Has[co+20]=1;

}

else if(NewNextB==1 && TypeBlock==3 && co%10>=3 && co/10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co-2]==0 && Has[co-10]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1-54, top-1, &blockred);

putimage(left-1, top-1-52, &blockred); putimage(left-1-108, top-1, &blockred);

Has[co]=Has[co-1]=Has[co-10]=Has[co-2]=1;

}

//第3种 反L型

else if(NewNextB==2 && (!co || TypeBlock==0 && co%10>=2 && co/10>=3 && Has[co]==0 && Has[co-1]==0 && Has[co-10]==0 && Has[co-20]==0))

{

if(!co){left=534; top=425; putimage(left-1, top-1, 49, 49, &blocks, 50, 0); }

else putimage(left-1, top-1, &blockred); putimage(left-1-54, top-1, &blockred);

putimage(left-1, top-1-52, &blockred); putimage(left-1, top-1-104, &blockred);

if(co){Has[co]=Has[co-1]=Has[co-10]=Has[co-20]=1; }

}

else if(NewNextB==2 && TypeBlock==1 && co%10<=6 && co/10>=2 && Has[co]==0 && Has[co-10]==0 && Has[co+1]==0 && Has[co+2]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1, top-1-52, &blockred); putimage(left-1+108, top-1, &blockred);

Has[co]=Has[co-10]=Has[co+1]=Has[co+2]=1;

}

else if(NewNextB==2 && TypeBlock==2 && co%10<=7 && co/10<=6 && Has[co]==0 && Has[co+1]==0 && Has[co+10]==0 && Has[co+20]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1, top-1+52, &blockred); putimage(left-1, top-1+104, &blockred);

Has[co]=Has[co+1]=Has[co+10]=Has[co+20]=1;

}

else if(NewNextB==2 && TypeBlock==3 && co%10>=3 && co/10<=7 && Has[co]==0 && Has[co-1]==0 && Has[co-2]==0 && Has[co+10]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1-54, top-1, &blockred);

putimage(left-1, top-1+52, &blockred); putimage(left-1-108, top-1, &blockred);

Has[co]=Has[co-1]=Has[co-2]=Has[co+10]=1;

}

//第4种 ⊥型

else if(NewNextB==3 && (!co || TypeBlock==0 && co/10>=2 && co%10<=7 && co%10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co+1]==0 && Has[co-10]==0))

{

if(!co){left=507; top=400; putimage(left-1, top-1, 49, 49, &blocks, 50, 0); }

else putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1-54, top-1, &blockred); putimage(left-1, top-1-52, &blockred);

if(co){Has[co]=Has[co-1]=Has[co+1]=Has[co-10]=1; }

}

else if(NewNextB==3 && TypeBlock==1 && co/10>=2 && co/10<=7 && co%10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co+10]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1, top-1+52, &blockred); putimage(left-1, top-1-52, &blockred);

Has[co]=Has[co+1]=Has[co-10]=Has[co+10]=1;

}

else if(NewNextB==3 && TypeBlock==2 && co/10<=7 && co%10<=7 && co%10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co+1]==0 && Has[co+10]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1-54, top-1, &blockred); putimage(left-1, top-1+52, &blockred);

Has[co]=Has[co-1]=Has[co+1]=Has[co+10]=1;

}

else if(NewNextB==3 && TypeBlock==3 && co/10>=2 && co/10<=7 && co%10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co+10]==0 && Has[co-10]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1, top-1+52, &blockred);

putimage(left-1-54, top-1, &blockred); putimage(left-1, top-1-52, &blockred);

Has[co]=Has[co-1]=Has[co+10]=Has[co-10]=1;

}

//第5种 右上楼梯型

else if(NewNextB==4 && (!co || (TypeBlock==0 || TypeBlock==2) && co%10<=7 && co%10>=2 && co/10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co-10]==0 && Has[co-9]==0))

{

if(!co){left=507; top=400; putimage(left-1, top-1, 49, 49, &blocks, 50, 0); }

else putimage(left-1, top-1, &blockred); putimage(left-1-54, top-1, &blockred);

putimage(left-1, top-1-52, &blockred); putimage(left-1+54, top-1-52, &blockred);

if(co){Has[co]=Has[co-1]=Has[co-10]=Has[co-9]=1; }

}

else if(NewNextB==4 && (TypeBlock==1 || TypeBlock==3) && co%10<=7 && co/10>=2 && co/10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co+11]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1, top-1-52, &blockred); putimage(left-1+54, top-1+52, &blockred);

Has[co]=Has[co+1]=Has[co-10]=Has[co+11]=1;

}

//第6种 左上楼梯型

else if(NewNextB==5 && (!co || (TypeBlock==0 || TypeBlock==2) && co%10<=7 && co%10>=2 && co/10>=2 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co-11]==0))

{

if(!co){left=507; top=400; putimage(left-1, top-1, 49, 49, &blocks, 50, 0); }

else putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1, top-1-52, &blockred); putimage(left-1-54, top-1-52, &blockred);

if(co){Has[co]=Has[co+1]=Has[co-10]=Has[co-11]=1; }

}

else if(NewNextB==5 && (TypeBlock==1 || TypeBlock==3) && co%10<=7 && co/10>=2 && co/10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co+10]==0 && Has[co-9]==0)

{

putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1, top-1+52, &blockred); putimage(left-1+54, top-1-52, &blockred);

Has[co]=Has[co+1]=Has[co+10]=Has[co-9]=1;

}

//第7种 四方块型

else if(NewNextB==6 && (!co || co/10<=7 && co%10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co+10]==0 && Has[co+11]==0))

{

if(!co){left=480; top=350; putimage(left-1, top-1, 49, 49, &blocks, 50, 0); }

else putimage(left-1, top-1, &blockred); putimage(left-1+54, top-1, &blockred);

putimage(left-1, top-1+52, &blockred); putimage(left-1+54, top-1+52, &blockred);

if(co){Has[co]=Has[co+1]=Has[co+10]=Has[co+11]=1; }

}

else if(NewNextB==7)//单位棋子

{

putimage(left-1, top-1, &blockred);

Has[co]=1;

}

else if(NewNextB==8 && Grey[co]==0)//消除单位棋子

{

putimage(left-1, top-1, 49, 49, &blocks, 0, 0);

Has[co]=0;

}

else if(NewNextB==9)//消除3x3棋子

{

putimage(left-1, top-1, 49, 49, &blocks, 0, 0); Grey[co]=Has[co]=0;

if(co/10>=2 && co%10>=2){putimage(left-1-54, top-1-52, 49, 49, &blocks, 0, 0); Grey[co-11]=Has[co-11]=0; }

if(co/10>=2) {putimage(left-1, top-1-52, 49, 49, &blocks, 0, 0); Grey[co-10]=Has[co-10]=0; }

if(co/10>=2 && co%10<=7){putimage(left-1+54, top-1-52, 49, 49, &blocks, 0, 0); Grey[co-9]=Has[co-9]=0; }

if(co%10>=2) {putimage(left-1-54, top-1, 49, 49, &blocks, 0, 0); Grey[co-1]=Has[co-1]=0; }

if(co%10<=7) {putimage(left-1+54, top-1, 49, 49, &blocks, 0, 0); Grey[co+1]=Has[co+1]=0; }

if(co/10<=7 && co%10>=2){putimage(left-1-54, top-1+52, 49, 49, &blocks, 0, 0); Grey[co+9]=Has[co+9]=0; }

if(co/10<=7) {putimage(left-1, top-1+52, 49, 49, &blocks, 0, 0); Grey[co+10]=Has[co+10]=0; }

if(co/10<=7 && co%10<=7){putimage(left-1+54, top-1+52, 49, 49, &blocks, 0, 0); Grey[co+11]=Has[co+11]=0; }

}

else

{

fillrectangle(-1, 499, 640, 520);

outtextxy(5, 502, "提示：不能放在这里！");

return 0;

}

return 1;

}

void BlockDel(bool Has[], bool Grey[], int \*score)

{

int i, j;

IMAGE blocks;

loadimage(&blocks, "rc\\blocks.jpg");

for(int ct=1; ct<=36; ct++)//从8x8开始往下降

{

int m, n;

if (ct== 1 || ct== 2 || ct== 5 || ct== 9 || ct==14 || ct==21)m=8;

else if(ct== 3 || ct== 4 || ct== 7 || ct==12 || ct==18 || ct==25)m=7;

else if(ct== 6 || ct== 8 || ct==11 || ct==16 || ct==23 || ct==28)m=6;

else if(ct==10 || ct==13 || ct==17 || ct==20 || ct==27 || ct==32)m=5;

else if(ct==15 || ct==19 || ct==24 || ct==29 || ct==31 || ct==34)m=4;

else if(ct==22 || ct==26 || ct==30 || ct==33 || ct==35 || ct==36)m=3;

if (ct== 1 || ct== 3 || ct== 6 || ct==10 || ct==15 || ct==22)n=8;

else if(ct== 2 || ct== 4 || ct== 8 || ct==13 || ct==19 || ct==26)n=7;

else if(ct== 5 || ct== 7 || ct==11 || ct==17 || ct==24 || ct==30)n=6;

else if(ct== 9 || ct==12 || ct==16 || ct==20 || ct==29 || ct==33)n=5;

else if(ct==14 || ct==18 || ct==23 || ct==27 || ct==31 || ct==35)n=4;

else if(ct==21 || ct==25 || ct==28 || ct==32 || ct==34 || ct==36)n=3;

for(i=1; i<=9-m; i++)//定位消除左上角格子

for(j=1; j<=9-n; j++)

{

int s, t;

for(s=i; s<i+m; s++)//判断在该m阶方阵是否全有

for(t=j; t<j+n; t++)

{

int co=10\*s+t;

if(!Has[co])goto cant;

}

s--; t--;

if(s==(i+m-1) && t==(j+n-1))

{

mciSendString("play score from 0", NULL, 0, NULL);

for(s=i; s<i+m; s++)

for(t=j; t<j+n; t++)

{

int co=10\*s+t;

int left=6+54\*(t-1), right=52+54\*(t-1), top=84+52\*(s-1), bottom=130+52\*(s-1);

settextstyle(20, 0, \_T("宋体")); settextcolor(WHITE);

int deltay=15;

while(--deltay){

fillrectangle(left, top, right, bottom);

outtextxy(left+15, top+deltay, "+1"); Sleep(8); }

if(!Grey[co]){

putimage(left-1, top-1, 49, 49, &blocks, 0, 0); Has[co]=0; }

else putimage(left-1, top-1, 49, 49, &blocks, 0, 50);

}

settextstyle(16, 0, \_T("宋体")); settextcolor(BLACK);

(\*score)+= m\*n;

}

cant:; }

}

}

int BlockCanPut(bool Has[], bool Grey[], int NewBlock)

{

int i, j;

for(int Type=0; Type<=3; Type++)//四种样式

for(i=1; i<=8; i++)

for(j=1; j<=8; j++)

{

int co=10\*i+j;

if(NewBlock==0 && (Type==0 || Type==2) && co%10<=5 && Has[co]==0 && Has[co+1]==0 && Has[co+2]==0 && Has[co+3]==0)return i\*10+j;

else if(NewBlock==0 && (Type==1 || Type==3) && co/10<=5 && Has[co]==0 && Has[co+10]==0 && Has[co+20]==0 && Has[co+30]==0)return i\*10+j;

else if(NewBlock==1 && Type==0 && co%10<=7 && co/10>=3 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co-20]==0)return i\*10+j;

else if(NewBlock==1 && Type==1 && co%10<=6 && co/10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co+2]==0 && Has[co+10]==0)return i\*10+j;

else if(NewBlock==1 && Type==2 && co%10>=2 && co/10<=6 && Has[co]==0 && Has[co-1]==0 && Has[co+10]==0 && Has[co+20]==0)return i\*10+j;

else if(NewBlock==1 && Type==3 && co%10>=3 && co/10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co-2]==0 && Has[co-10]==0)return i\*10+j;

else if(NewBlock==2 && Type==0 && co%10>=2 && co/10>=3 && Has[co]==0 && Has[co-1]==0 && Has[co-10]==0 && Has[co-20]==0)return i\*10+j;

else if(NewBlock==2 && Type==1 && co%10<=6 && co/10>=2 && Has[co]==0 && Has[co-10]==0 && Has[co+1]==0 && Has[co+2]==0)return i\*10+j;

else if(NewBlock==2 && Type==2 && co%10<=7 && co/10<=6 && Has[co]==0 && Has[co+1]==0 && Has[co+10]==0 && Has[co+20]==0)return i\*10+j;

else if(NewBlock==2 && Type==3 && co%10>=3 && co/10<=7 && Has[co]==0 && Has[co-1]==0 && Has[co-2]==0 && Has[co+10]==0)return i\*10+j;

else if(NewBlock==3 && Type==0 && co/10>=2 && co%10<=7 && co%10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co+1]==0 && Has[co-10]==0)return i\*10+j;

else if(NewBlock==3 && Type==1 && co/10>=2 && co/10<=7 && co%10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co+10]==0)return i\*10+j;

else if(NewBlock==3 && Type==2 && co/10<=7 && co%10<=7 && co%10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co+1]==0 && Has[co+10]==0)return i\*10+j;

else if(NewBlock==3 && Type==3 && co/10>=2 && co/10<=7 && co%10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co+10]==0 && Has[co-10]==0)return i\*10+j;

else if(NewBlock==4 && (Type==0 || Type==2) && co%10<=7 && co%10>=2 && co/10>=2 && Has[co]==0 && Has[co-1]==0 && Has[co-10]==0 && Has[co-9]==0)return i\*10+j;

else if(NewBlock==4 && (Type==1 || Type==3) && co%10<=7 && co/10>=2 && co/10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co+11]==0)return i\*10+j;

else if(NewBlock==5 && (Type==0 || Type==2) && co%10<=7 && co%10>=2 && co/10>=2 && Has[co]==0 && Has[co+1]==0 && Has[co-10]==0 && Has[co-11]==0)return i\*10+j;

else if(NewBlock==5 && (Type==1 || Type==3) && co%10<=7 && co/10>=2 && co/10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co+10]==0 && Has[co-9]==0)return i\*10+j;

else if(NewBlock==6 && co/10<=7 && co%10<=7 && Has[co]==0 && Has[co+1]==0 && Has[co+10]==0 && Has[co+11]==0)return i\*10+j;

else if(NewBlock==7 && Has[co]==0 || NewBlock==8 || NewBlock==9)return i\*10+j;

}

mciSendString("play dead from 0", NULL, 0, NULL);

return 0;

}