程序名称：博弈五子棋

作者：软工1901苑明飞 191203123

开发环境：Visual Studio 2017

图形库：EasyX

最后修改：2020年3月29日

游戏采用c语言加c++图形库的方式实现，游戏纯按键操作，可视化菜单，有音效、背景音乐和贴图。游戏分为两种模式，双人模式和人机模式。双人模式中可以设置棋手昵称和先手方，未加入背景颜色选择功能的原因是其他背景颜色黑白棋子颜色显示效果不好。

\*\*\*主菜单操作：键盘右下角方向箭头进行选择，回车键确认，游戏纯按键操作。\*\*\*

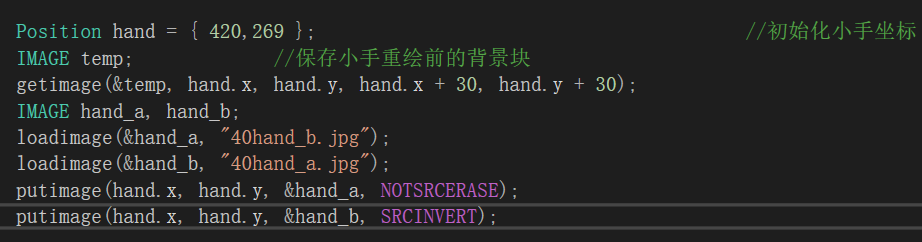
\*\*\*游戏资源文件必须与源代码文件放在同一个文件夹中才可以正常显示贴图和音乐。\*\*\*

下面分别介绍主要实现的逻辑和算法：

1.菜单部分:（坐标采用结构体的方式储存）

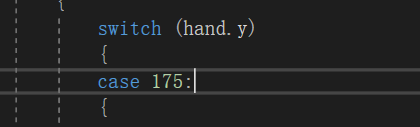
* 1. 小手指针的上下移动：

通过按键改变其坐标，用保存的背景图覆盖原图并在新坐标位置重绘的方式实现动画效果。（由于easyx图形库不支持png格式的透明图片，故采用位运算的方式过滤颜色实现透明效果）



* 1. 界面的选择：

用了switch语句对小手的y坐标进行判断，当y的坐标在特定位置按下回车键时执行特定的功能代码。

。

* 1. 返回主菜单：

使用goto语句在同一作用域内跳转。

# 2.绘图部分：

# 五子棋坐标图

游戏棋盘部分570 \* 570像素 ,游戏整个窗口是750 \* 570 像素。

本游戏动画效果均采用重绘加遮盖原图的方式实现，没有使用清屏函数，避免闪屏。

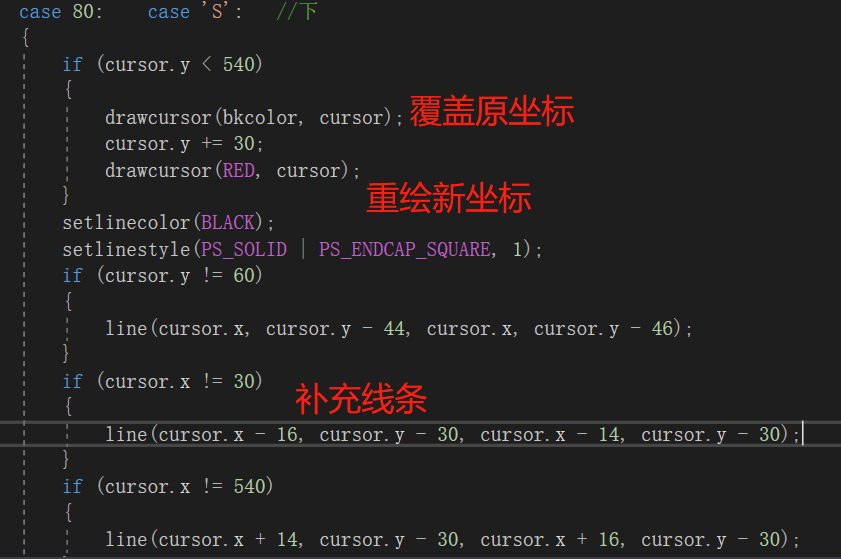
双人模式：绘制棋盘，绘制执子方，绘制模式，绘制光标，绘制昵称。

人机模式：绘制棋盘，绘制电脑白棋，绘制玩家黑棋，绘制光标，绘制模式。

3.游戏主体：（坐标采用结构体的方式储存）

(1) 光标的移动：

死循环监听键盘消息，接收到指定方向消息，判断下一位置是否在棋盘内，如果在，用背景图覆盖原光标，在新坐标位置重新绘制，光标坐标更新。（此处有个问题：因为棋盘是用画线函数画的，在覆盖原光标的时候会破坏线条，所以在重绘之后，还需要进行补线条）



1. 落子：

·双人模式：

光标坐标位置绘制棋子（绘制棋子函数进行判断当前坐标是否有棋子，绘制后，切换执子方，总棋子数加1），重绘执子方，进行胜负产生判断，如果产生胜负，则调用结束游戏函数（该函数根据当前执子方进行输赢显示，同时让玩家选择是否重新开始还是返回菜单），如果未产生胜负，则进行平局判断，如果平局则用不同的参数调用关闭游戏函数（提示玩家游戏平局，让玩家选择重新开始还是返回菜单）。如果平局也没有产生，则继续游戏。

·人机模式：

玩家先手，电脑白子，玩家黑子。人机模式相比双人模式多了一次胜负和平局判断，在玩家落子后，进行胜负和平局判断，通过计算获取对电脑最有利的坐标，电脑落子，再次进行胜负和平局判断。

1. ESC键：

在游戏中返回主菜单，执行和主函数同样的内容。

# 判断胜负算法：

棋盘分为四个方向：横，竖，左斜，右斜。

判断原理：各个方向的棋子数大于等于5就产生胜负。

实现过程：使用循环在落子位置向上下（其他方向同理）两个方向寻找同色棋子（此处使用getpixel函数进行颜色比对），找到一个number加1，遇到非同色或者是棋盘空位置，则break跳出循环。因为有光标位置棋子，所以只要number >= 4即可产生胜负。

判断平局：

平局的判断是根据总棋子数量，当总棋子数量占满整个棋盘，并未产生胜负，则平局。

# 人机寻找最优位置算法：

原理：玩家落子后，人机对棋盘的每一个空位置进行评分，评分越高的点对人机越有利。

评分有两个方面：

1. 下这个棋子后人机可以连成几子，连成数量越多，得分越高。
2. 下这个棋子后，可以打断玩家连成几子，打断连成数量越多，得分越高。

这两个得分之和就是最终得分，哪个最终得分最高，电脑就在此处落子。

五子棋的摆法主要分以下几种：

A：棋子 X：棋盘空位置

成五：AAAAA

活四：XAAAAX

死四：XAAAA或AAAAX

活三：XAAAX

死三：XAAA或AAAX

活二：XAAX

死二：XAA或AAX

活一：A

死一：XA或AX

计分表：

|  |  |  |  |
| --- | --- | --- | --- |
| 电脑连成五子 | 100000 | 阻碍玩家连成五子 | 1000 |
| 电脑连成活四 | 200 | 阻碍玩家连成活四 | 100 |
| 电脑连成死四 | 50 | 阻碍玩家连成死四 | 20 |
| 电脑连成活三 | 30 | 阻碍玩家连成活三 | 10 |
| 电脑连成死三 | 8 | 阻碍玩家连成死三 | 5 |
| 电脑连成活二 | 2 | 阻碍玩家连成活二 | 1 |
| 电脑连成死二 | 2 | 阻碍玩家连成死二 | 1 |
| 电脑连成活一 | 1 | 阻碍玩家连成活一 | 0 |
| 电脑连成死一 | 1 | 阻碍玩家连成死一 | 0 |

具体实现：

先定义一个可以存下所有坐标的二维数组，在计分函数中遍历整个棋盘，寻找空位子，再用循环检索该空位置的八个方向上落子情况(getpixel函数)，进行相应的打分。由此获得每个空位子的分数。再用寻找分数最高函数找出分数最高的坐标位置进行落子。具体请看源文件的c\_downchess函数和getmax函数。

**7.游戏整体流程思维导图：**(也已文件形式一起打包)

