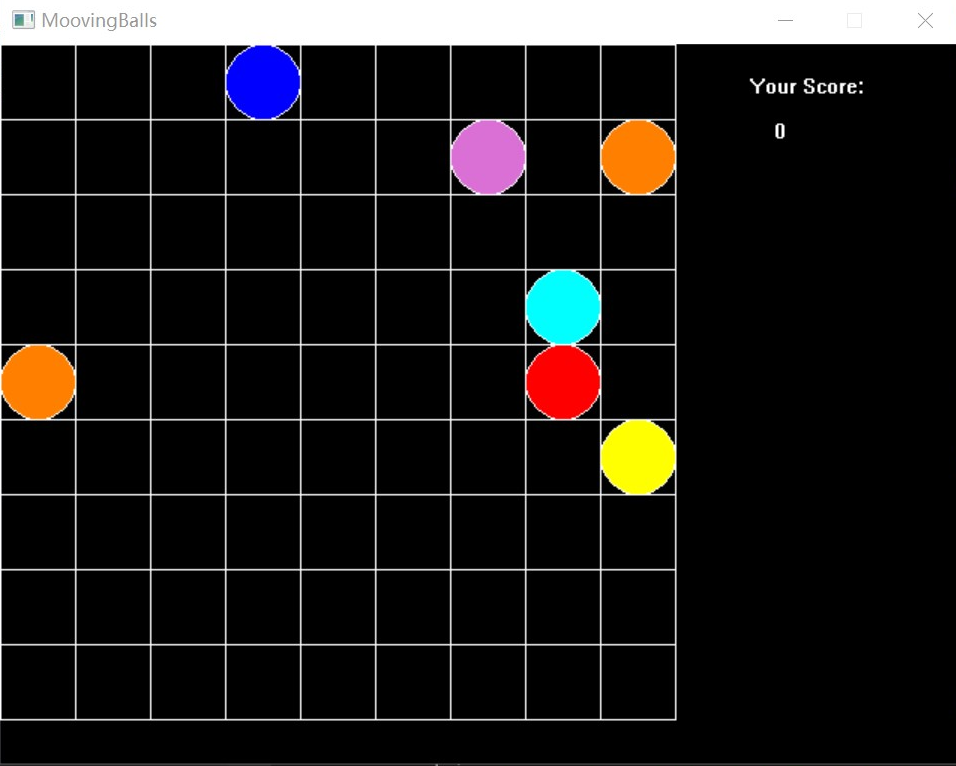
**五子连珠游戏程序设计说明**

梁厚（学号：2051840）

【游戏概述】

棋盘大小是9X9，会出现6种颜色的珠子。初始状态棋盘上7个随机位置分布着7个随机颜色的珠子。通过移动珠子将同色的珠子连在一起来消除后得分。当无珠子可以移动时程序结束。



1. 程序大致结构
2. 程序实现方案概述
   1. 数据存放

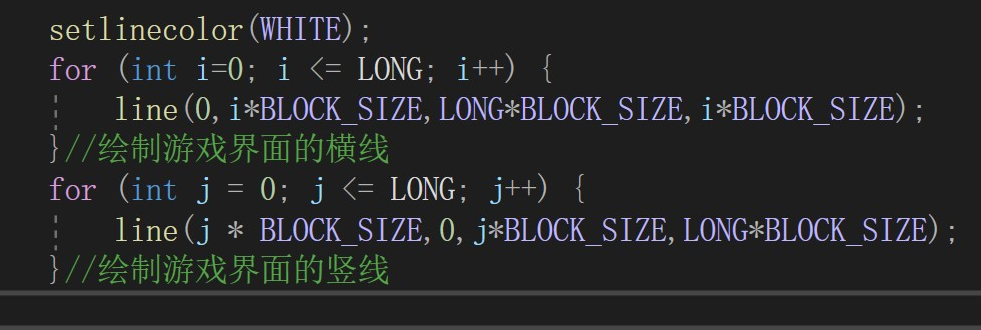
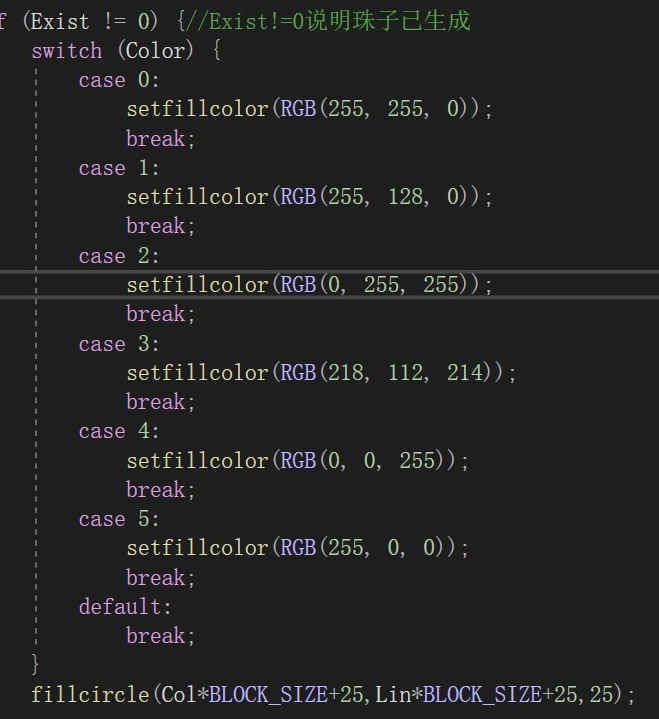
定义一个Ball类表示每一位置上的珠子。用全局对象数组Ball[LONG][LONG](LONG在头文件中宏定义为9，表示棋盘的长、宽)记录与棋盘上的珠子有关的所有游戏数据，全局整形变量score记录玩家的得分。

其中，Ball类的声明及定义如下：

说明：每个位置上的珠子都有一个对应的Ball类对象。都有行列、颜色等数据信息。而在输出屏上该位置是否有珠子，取决于对象的bool类型Exist的数据成员。若珠子应在屏幕上显示，则Exist为true（珠子存在），反之则为false（不存在）。

* 1. 图形化

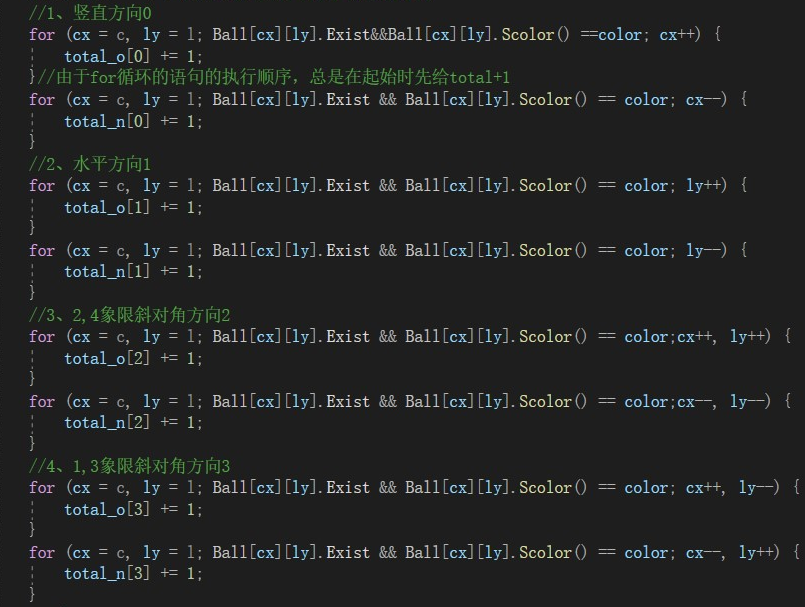
下载了EasyX的插件，调用graphics.h图形库里的相关函数实现图形化过程。

1. 棋盘绘制：调用图形库里的Line函数绘制网格的横线和竖线
2. 珠子绘制：调用图形库里的setfillcolor(在图形内填充颜色)、fillcircle(绘制圆)。不同颜色的珠子填充不同的颜色，颜色使用RGB()函数设置。
3. 玩家当前得分通过outtextxy()函数输出到显示屏。
   1. 内部逻辑功能
      1. 检查五子连珠并消去

假定原来的游戏格局无五颗以上连在一起的珠子，则唯一变动的只有玩家移动的那颗，所以游戏格局发生改变后，出现五颗珠子连在一起情况的只可能是玩家移动的珠子附近。Check()函数负责检查是否有五子（或五子以上）连珠的情况。具体的实现思路是：

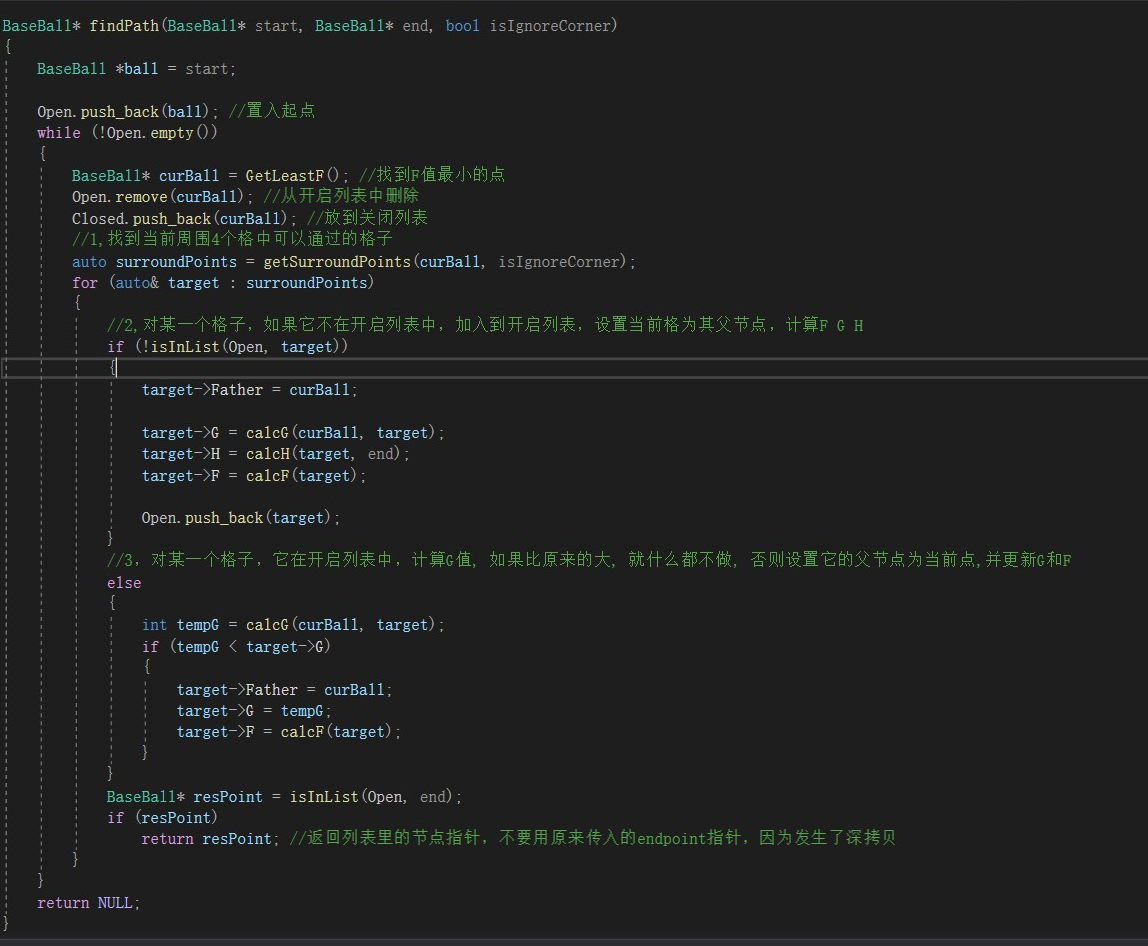
1. 以玩家移动的珠子为原点，记为Ball[c][l]；
2. 用for()循环检索每个方向上与Ball[c][l]颜色相同的珠子的个数，并用一个数组记录下这组数字；
3. 每个方向都有相应的编号。通过for()循环找到五子连珠的方向，返回这个方向的编号以及这个方向上两边的颜色相同的珠子数量（五子连珠的方向可能有多个）
4. 每检测到一个符合条件的方向就调用ClearBall()函数修改相应位置上Ball对象的参数（即Ball对象的Exist数据成员变为false，珠子转变为不存在）

考虑到系统每次生成的3颗珠子也可能与原来的珠子形成五子连珠的情况，所以对新生成的3颗珠子也同样调用Check()函数。设置一个jud参数区别是玩家连成的五子连珠，还是系统连成的五子连珠，从而判断是否应该加上相应的分数。



* + 1. 判断路径是否合理

使用了A\*算法。



1. 小结

整个程序通过全局Ball类对象数组和全局整形变量score记录所有的游戏数据，通过graphics.h图形库中的各个函数实现游戏的图形化过程，通过自己编写的一系列承担相应的功能和任务函数，最终实现五子连珠的控制台小游戏。