俄罗斯方块Tetris（2018.6.5）

编写程序之前的知识储备：

首先设计控制台窗口

设计console类和window类（console是window的成员），这两个类应该具备以下能力：

控制窗口的标题，窗口的大小，缓冲区大小，光标

完全控制输出的能力，包括但不限于文字的显示，颜色，前景色和背景色

有关的现成的函数：

? GetStdHandle() // 获得句柄

? SetConsoleCursorInfo() // 设置光标信息

? SetConsoleWindowInfo() // 设置窗口信息

? SetConsoleScreenBufferSize() //设置窗口缓冲区大小

? SetConsoleTitle() //设置标题

? WriteConsoleOutputCharacter() 和 WriteConsoleOutputAttribute() //控制输出的函数

最后这两个函数至关重要，承担了游戏的打印(渲染)任务。

相关资料查询如下：

// 函数原型：

BOOL WriteConsoleOutputCharacter( // 在指定位置处插入指定数量的字符

HANDLE hConsoleOutput, // 句柄

LPCTSTR lpCharacter, // 字符串

DWORD nLength, // 字符个数

COORD dwWriteCoord, // 起始位置

LPDWORD lpNumberOfCharsWritten // 已写个数 );

/\* 参数简介：

hConsoleOutput：控制台输出句柄，通过调用GetStdHandle函数获得

HANDLE hnd;

hnd=GetStdHandle(STD\_INPUT\_HANDLE);

lpCharacter：要输出的字符串

nLength：输出长度

dwWriteCoord：起始位置

pNumberOfCharsWritten：已写个数，通常置为NULL 其中，COORD是个结构体变量类型\*/ typedef struct \_COORD { SHORT X; SHORT Y; } COORD;

游戏逻辑设计

游戏中方块的的种类，有7种，分别是“L”、反“L”、“|”、“土”、“■”、“Z”，反“Z”。每一种形状有4中变形，分别是四个方位的旋转

每种形状存二维数组中。由于一种有7种形状，4种变形，所以可以定义数组为，A[7][4][4][4]

关于方块的变形，方块逆时针旋转，由于在游戏中已经将方块的变形列出来了，所以这里只需要改变数组a[types][direction][4][4]中的方向值。这里用(m\_currentDir + 1) % 4，因为方向值为0-3，这种方法可以保证其在0—3之间循环。注：在变形过程中需要实时更新空白处以及变形处的颜色。以免被覆盖。（着色原理为：判断此处是否有值，根据实时变化的坐标（此坐标一般指4\*4方块左上角的坐标），计算出有值处的坐标，然后将此处变为空白。变形之后同样执行这段代码，只是将空白变为有颜色。代码如下：

if (bk[m\_currentDiamonds][m\_currentDir][i][j] == 1)