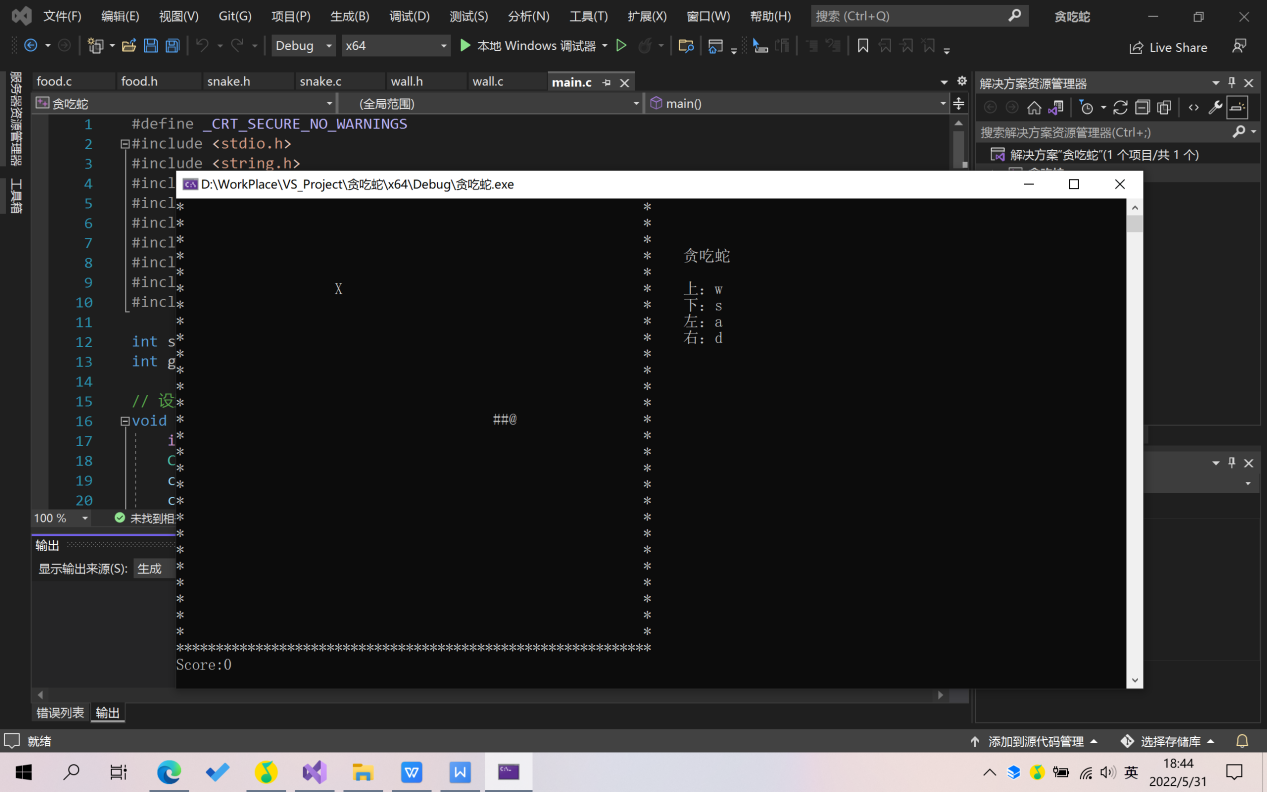
# C语言练手项目——贪吃蛇

#### 项目简介

本项目使用C语言编写了一个在控制台运行的贪吃蛇程序，玩家通过操控键盘的“wasd”四个按键实现对“蛇”身体上下左右四个方向的控制，同时具有显示玩家得分的功能，程序运行截图如下：



#### **程序实现**

##### 墙体

墙体是本程序的构成部分之一，充当了作为判断游戏结束条件的重要角色。将墙体的相关信息和函数单独创建相应的头文件和源文件。在头文件中，定义了宽度和高度两个常量，再定义了一个函数初始化墙体；在源文件中，对初始化墙体进行了实现：遍历宽高，判断该位置为边界则打印“\*”，分则打印空格。

##### 蛇

蛇是本程序中用户控制的主体，本程序中使用结构体封装了蛇的相关信息：

// 蛇每一节的结构体

struct Body

{

int x;

int y;

};

// 整个蛇的结构体

struct Snake

{

struct Body body[(HEIGHT - 2) \* (WIDTH - 2)];

int size;

}snake; // 蛇结构体变量

初始化Snake时封装了body结构体数组，数组大小为墙体围成的最大面积。在源文件中，添加了初始化蛇的函数，初始化蛇长度为3，位置为屏幕中央，方向向右。

完成了蛇的初始化后，将该函数添加到main函数中执行，此时还不能显示出来，需要通过函数updata()打印。

##### **食物**

食物需要随机刷新在墙体之内除了蛇身的位置，需要在初始化的时候进行判断。此外，需要在main函数中加入随机数种子，以确保真随机。

##### 蛇的移动和增长

蛇移动的逻辑为：蛇身（头部除外）第i节每次移动到前一时刻i-1节所在的位置；蛇头向键盘输入的方向移动；蛇尾的位置打印为空格。

蛇的增长逻辑为：蛇头和食物碰撞后增加蛇的长度，然后调用updata函数打印即可。

##### **游戏结束**

当蛇头碰到墙壁或者蛇头碰到自身时结束游戏。由于撞自身打印时会不显示蛇头，因此需要判断结束的种类以增加蛇头打印。

##### **其他函数**

本程序用到了隐藏控制台光标和设置控制台光标位置的函数。