贪吃蛇

——**初期分析**

1. 蛇：考虑到每个蛇节点包含x，y坐标以及指向下一个节点的指针（坐标用c语言内置类型），用结构体类型来表示每一个蛇身节点；
2. 坐标：c语言有内置的坐标结构体类型，包括x和y坐标，一个坐标对应一个字符的大小，一个方形x方向占两个字符，即两个x坐标，y方向占一个；
3. 光标位置，用一个Pos函数来移动光标到指定位置完成打印输出；
4. 生成小蛇；
5. 随机出现食物：食物的出现不能超出坐标边界，使用srand和rand方法生成随机食物；在生成食物的时候注意由于一开始的x坐标出现在偶数上，所以rand得到的食物的x坐标也必须是偶数，否则蛇吃不到，y坐标没有限制；
6. 蛇移动，寻找食物：程序的核心，每次蛇的移动都是在响应一个移动事件，在每两次移动事件之间加上时间延缓来使蛇的移动看起来自然。当然可以通过加速和减速功能来减小或者增加延缓时间。如果蛇吃到食物，把食物节点作为新的头节点，后面的节点位置不动。否则把食物作为头节点，后面的节点依次前移，末尾节点设为空并刷新。
7. 碰墙，碰自己，用户返回，游戏结束。

——**几大模块**

1. 初始化模块：

功能分析：

初始化欢迎界面：

初始化地图

初始化蛇身

具体实现：

初始化欢迎界面：定位好坐标并打印相关欢迎信息即可；

初始化地图：定位好坐标并依次打印出方块即可；

初始化蛇身：先生成一个节点，再自左向右依次设置剩下的节点，最后遍历打印出节点（方块）；

二：判断模块：

功能分析：

判断是否碰墙

判断是否碰到自己

判断游戏是否结束（通过标志位）

具体实现：

判断是否碰墙：如果蛇头坐标超出地图范围，判定为撞墙，修改游戏结束标志位；

判断是否碰到自己：从头部开始遍历蛇，如果哪个节点坐标和头部节点坐标重合，就判断撞到自己；

判断游戏是否结束（通过标志位）：根据标志位的状态判断游戏是否结束；

三：游戏模块：

功能分析：

随机出现食物

玩家操作蛇移动，每次响应一个输入事件

游戏结束

具体实现：

随机出现食物：每次通过rand函数生成一个随机坐标，把x转化为偶数坐标，y不变，然后把光标移动到对应位置打印出食物；

玩家操作蛇移动，每次响应一个键盘输入事件：游戏核心，每次响应移动事件后判断是否吃到食物：

1. 吃到食物：把食物节点作为新的头节点，其余节点位置不变，游戏分数加point；
2. 没吃到食物，也把食物节点作为新的头节点，剩余节点依次向前移动；这样原来的最后一个节点就成为了空节点，打印出空格来覆盖原来的方块。

游戏结束：结束游戏。