

Project-3 实验报告

Task1:贪吃蛇的游走程序

```
//输出字符矩阵
//WHILE not 游戏结束 DO
// ch=等待输入
// CASE ch DO
//   'A':左前进一步 {
//     蛇头横坐标-1, 纵坐标不变;
//     将蛇头原来的坐标赋给第一截身体;
//     将第一截身体原来坐标赋给第二截身体;
//     .....
//     将第 n-1 截身体原来坐标赋值给第 n 截身体;
//   }
//   break;
//   'D':右前进一步 {
//     蛇头横坐标+1, 纵坐标不变;
//     将蛇头原来的坐标赋给第一截身体;
//     将第一截身体原来坐标赋给第二截身体;
//     .....
//     将第 n-1 截身体原来坐标赋值给第 n 截身体;
//   }
//   break;
//   'W':上前进一步 {
//     蛇头横坐标不变, 纵坐标-1;
//     将蛇头原来的坐标赋给第一截身体;
//     将第一截身体原来坐标赋给第二截身体;
//     .....
//     将第 n-1 截身体原来坐标赋值给第 n 截身体;
//   }
//   break;
//   'S':下前进一步, break {
//     蛇头横坐标不变, 纵坐标+1;
//     将蛇头原来的坐标赋给第一截身体;
//     将第一截身体原来坐标赋给第二截身体;
//     .....
//     将第 n-1 截身体原来坐标赋值给第 n 截身体;
//   } break;
// END CASE
// 输出字符矩阵
//END WHILE
//输出 Game Over!!!
```

Task2: 编写程序, 使得 snake 能吃一个食物就长一节
写 [编写 actions 伪代码]

Snake | Head on Wall & Body （参见游戏事件编程）

CALL gameover

Snake | Head on Blank

CALL snakeMove

Snake | Head on Food

CALL snakeMove

snakeLength++

SET the coordinate of last body AS the coordinate of original last body

CALL put_money

Reprint the map

Task3: [选择性任务]编写人工智能程序，使得 snake 每秒自动走一步。

[程序要求]:

（1）请修改初始化字符矩阵，或者通过文件读入地图。地图中有一些你设定的障碍物（墙）

（2）请实现如下算法决定蛇行走的方向：

```
function whereGoNext(Hx,Hy,Fx,Fy) {  
  //movable[3]={“a”,“d”,“w”,“s”} 记录可走的方向  
  //用数组 distance[3]={0,0,0,0} 记录离食物的距离  
  //分别计算蛇头周边四个位置到食物的距离。  
  // 返回 movable[p]  
}
```

（3）一个长度为 5 的障碍物能困死该自动跑的蛇吗？
能。当这个障碍物恰好把蛇头套住的时候。

下面我呕心沥血改进的介绍，很重要，一定要看！

问题	改进
贪心算法会走进死胡同	改为深度优先搜索算法
深度优先搜索算法向每一个方向搜索太耗资源，也慢	深搜结合贪心，每一步搜索都优先向最靠近食物的方向搜索
每走一步搜索一次，当地图大且障碍物多时候效率极其低下	设立一个 route 数组记录走到食物所需的步骤，这样只要深搜一次找到了，便可沿着走，吃到食物。
有时候，当蛇身横跨整个地图，将蛇头和食物隔开，搜索不到	设立一个 restLength 变量，搜索的时候模拟走路，在 Restlength <MAN_restLength 时，每搜索一个格就把尾巴缩进来一个格，让出位置。
其他	
不想每次改地图都慢慢敲	写了个地图生成程序，用了个宏定义 MAP_SIZE 控制大小
	随机生成障碍物
	进入时选择模式，0 是手动，1 是自动

