# 运作流程

玩家通过移动贪食蛇，吃掉场上的“食物”，可以使得蛇身增加以及分数增加，在普通模式中可以穿过游戏边界到达另一边界，在竞技模式中可以比拼，一方率先碰到另一方，则前者判定输。

# 模式

* 经典模式：

默认旋转方式：只能水平或者竖直九十度旋转

默认移动方式：沿着自身的正方向Y轴移动

游戏边界：不能穿过

死亡：碰到边缘，蛇头碰到蛇身

* 普通模式：

默认旋转方式：可以向任意方向平滑旋转

默认移动方式：沿着自身的正方向Y轴平滑移动（使用插值或者每次移动的距离变小，移动的频率变高达到看似平滑目的）

游戏边界：能穿过

死亡：蛇头碰到蛇身

* 竞技模式：

穿墙：能穿过边缘

死亡：蛇头碰到蛇身

待开发~~~~~~~~~~~

# 操作方式

PC： 移动：上下左右=》wsad ||↑↓←→

加速移动：长按移动键1秒||空格

移动端：移动：EasyTouch 加速移动：Button

# 移动的实现原理

## 方式1：

**蛇头每次移动，每个蛇身（节点）都继承上一个蛇身（节点）的位置。**

* **蛇头移动**：具体实现方式：转弯逻辑修改蛇头的Rotation局部坐标，Y轴一直指向蛇头的正前方，之后不断让蛇头向Y轴前进。
* **蛇身移动**：当前蛇身节点只需要在前一个蛇身节点位置即可，不需要自己移动，转向
* **坑**

坑1：蛇身节点移动时候要从最后一个往前移动

坑2：蛇身不可以向蛇头移动方式一样（蛇身与蛇头转向一致，蛇身自己向自己的Y轴前进），否则会出现蛇头正方向指向身体现象

坑3：蛇身的密度跟蛇头每次移动的距离有关

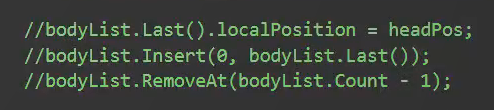
坑4：上一次蛇头的位置与当前蛇头的位置容易混淆

## 方式2：

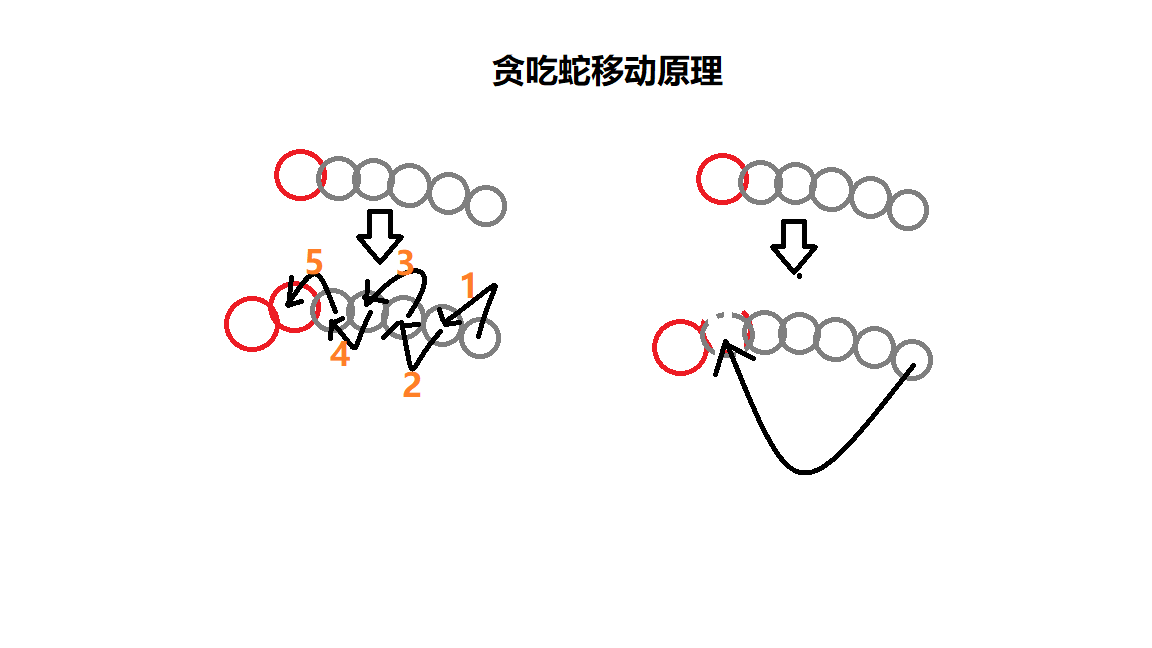
**蛇头每次移动，蛇身末尾最后一个（节点）直接移至蛇头移动之前的位置即可，但是这种方式在双色蛇身上改变贴图颜色会被教繁琐。**

* 具体实现：

把蛇身最后一个节点坐标改为蛇头移动之前的位置，再新增一个List索引放在之前第一个蛇身节点之前位置，最后再删除最后一个List索引



## 原理图



# 具体实现

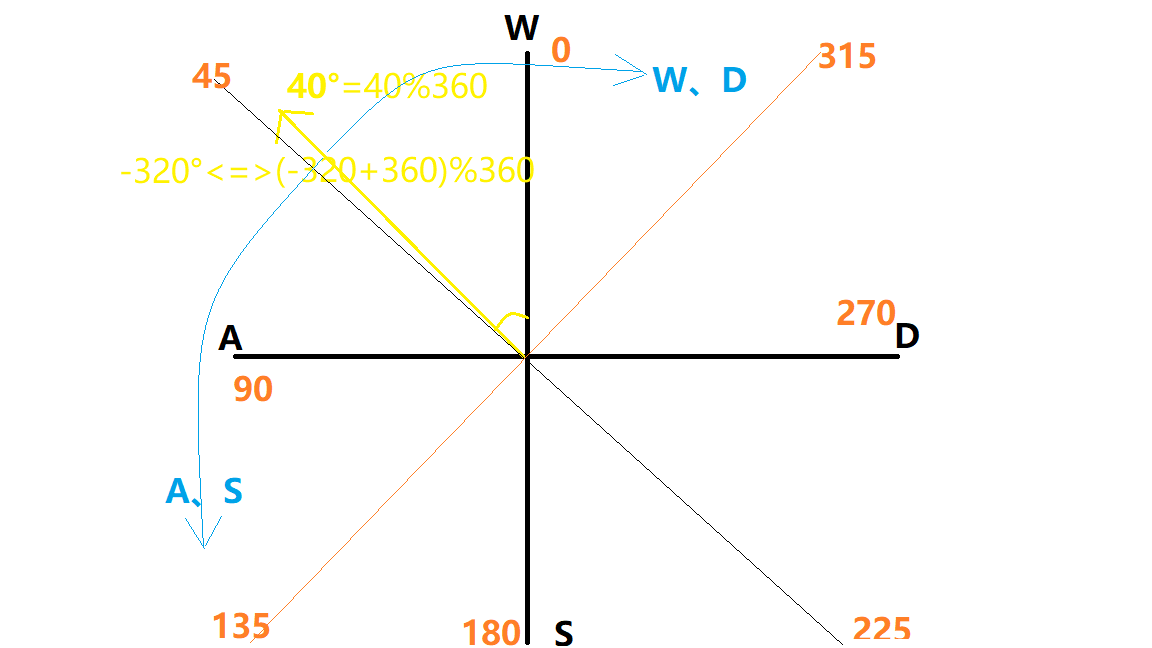
## 任意方向旋转

普通模式下蛇头可向任意方向旋转具体形式有以下两种方式：

### 方式1：

* 根据当前角度-》玩家输入WASD-》当前角度偏移

假定当前移动的方向为世界坐标系的40°方向 ,则W、D往右侧偏移，A、S往左侧偏移。



如图可得伪代码 令当前角度为x

If(x>=0 && x<=90)

W、D=》 x--;

A、S =》 x++;

Else if(x>90 && x<=180) x--;

A、W=》 x--;

S、D =》 x++;

Else if(x>180 && x<=270) x--;

A、S =》 x--;

D、W =》 x++;

Else if(x>270 && x<360) x--;

D、S=》 x--;

A、W =》 x++;

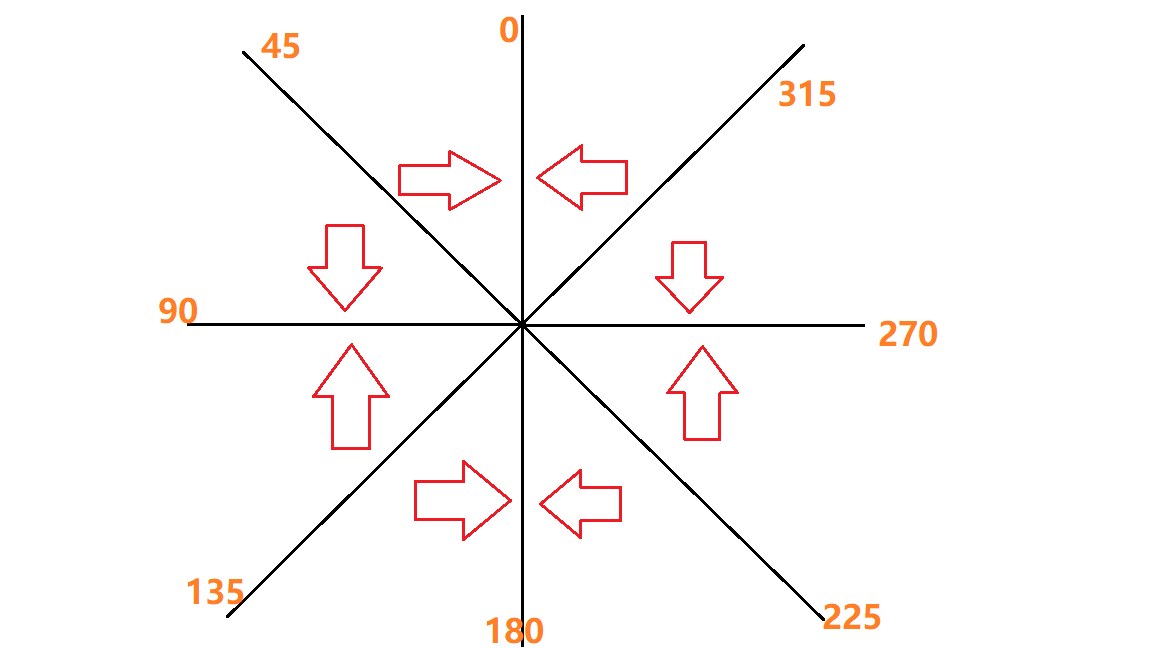
Else Debug.LogError(“...”)

### 坑：

## 每次旋转的偏移量不要用浮点型，一定要用整形，因为使用前者会发生数据精度越界，会出现bug或者死循环

### 原理2：

* **根据玩家输入WASD-》获取当前角度-》当前角度偏移**



## 蛇头蛇身碰撞Bug：

**描述：**

蛇头与蛇身的碰撞事件，之前为了避免蛇头移动过程中直接回头发生碰撞，设计了不允许往反方向移动的逻辑(想要移动的方向与当前移动的方向做匹配)，但是测试中，例如当前向上移动，突然向右移动、又突然向下移动，结果直接发生了蛇头直接回头碰撞蛇身的状况。

**原因：**

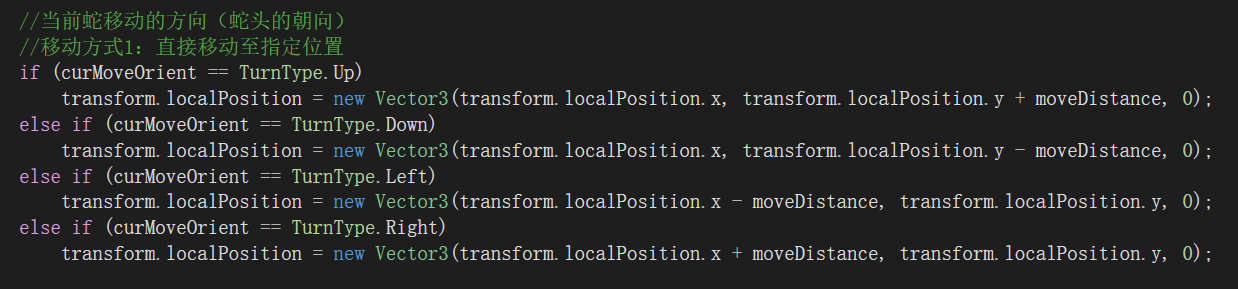
向右移动+向下移动所用的时间 < 蛇的移动频率 ，还没来得及执行上一个移动指令，又有了第二条指令，越过了上一个移动指令直接执行了第二天指令。

**解决方案：**

设置bool类型变量控制是否接收玩家输入，确保在蛇完成当前移动指定后方可接收下一按键指令。

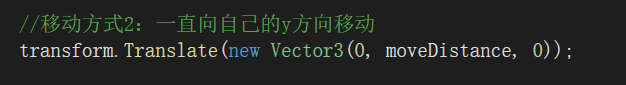
## 蛇头移动

### 移动方式1：直接移动至指定位置



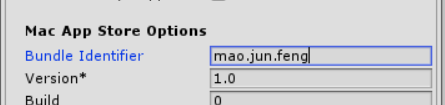
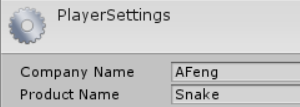
### 移动方式2：一直向自身Y轴方向移动

**注意：这种方式一定要配合游戏物体旋转使用**

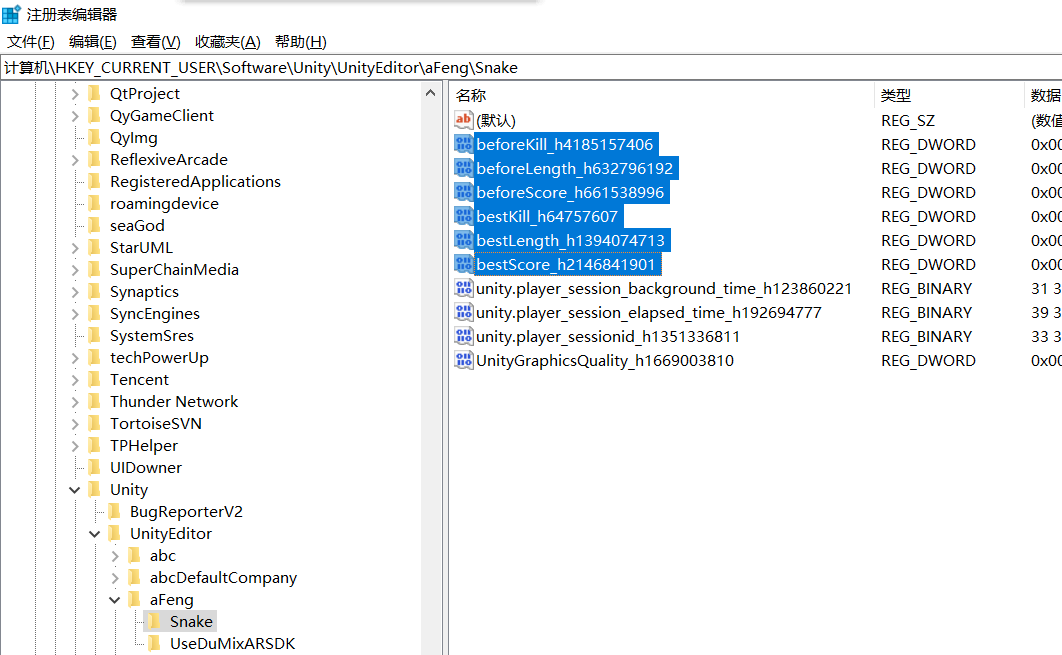


# 6.存档注册表

成绩的注册表：



计算机\HKEY\_CURRENT\_USER\Software\Unity\UnityEditor\aFeng\Snake



# 7.其他

