**贪吃蛇开发日志**

# 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：

旋转方式：水平或者竖直90°旋转，自由360°旋转

默认移动方式：沿着自身的正方向Y轴移动

游戏边界：不允许穿过边界

死亡：蛇头碰到边缘，蛇头碰到蛇身

* 普通模式：

旋转方式：水平或者竖直90°旋转，自由360°旋转

默认移动方式：沿着自身的正方向Y轴平滑移动

游戏边界：无边界

死亡：蛇头碰到边缘，蛇头碰到蛇身

* 竞技模式：

旋转方式：水平或者竖直90°旋转，自由360°旋转

默认移动方式：沿着自身的正方向Y轴平滑移动

游戏边界：不允许穿过边界

死亡：1.玩家蛇头碰到敌方的蛇头、蛇身或者边界

胜利：敌方蛇头碰到玩家的蛇头、蛇身

# 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

持续加速：前者Button上滑经过持续加速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

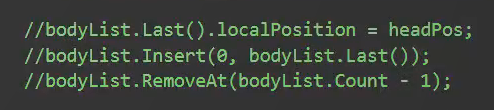
坑4：上一次蛇头的位置与当前蛇头的位置容易混淆

## 方式2：

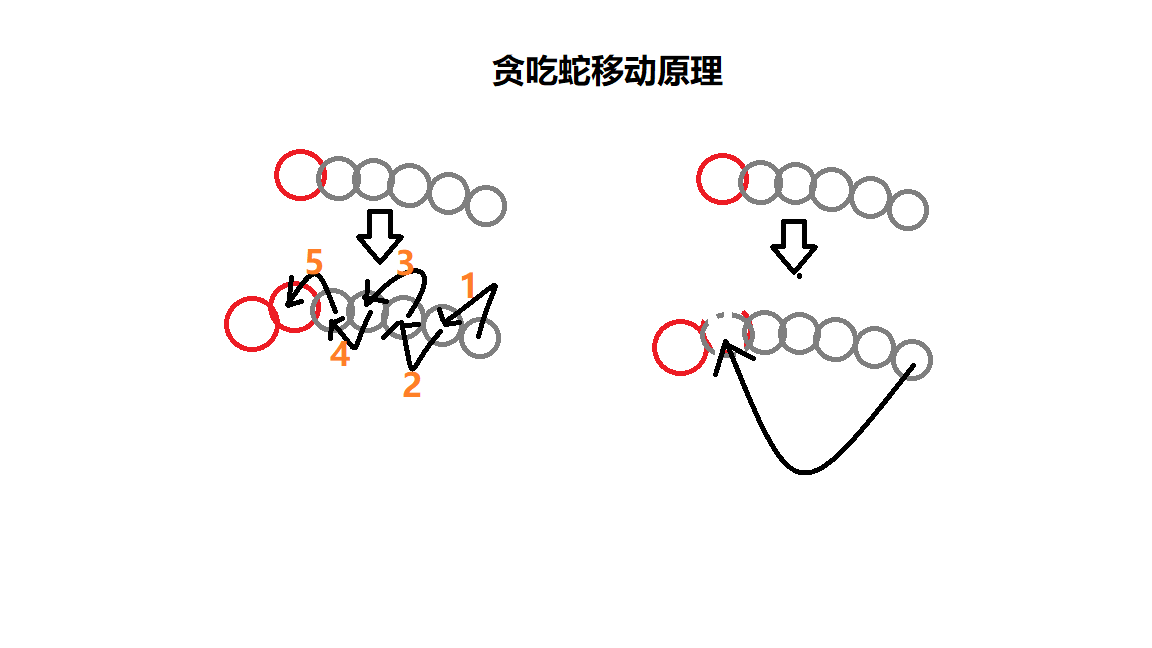
**蛇头每次移动，蛇身末尾最后一个（节点）直接移至蛇头移动之前的位置即可，但是这种方式在双色蛇身上改变贴图颜色会被教繁琐。**

* 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

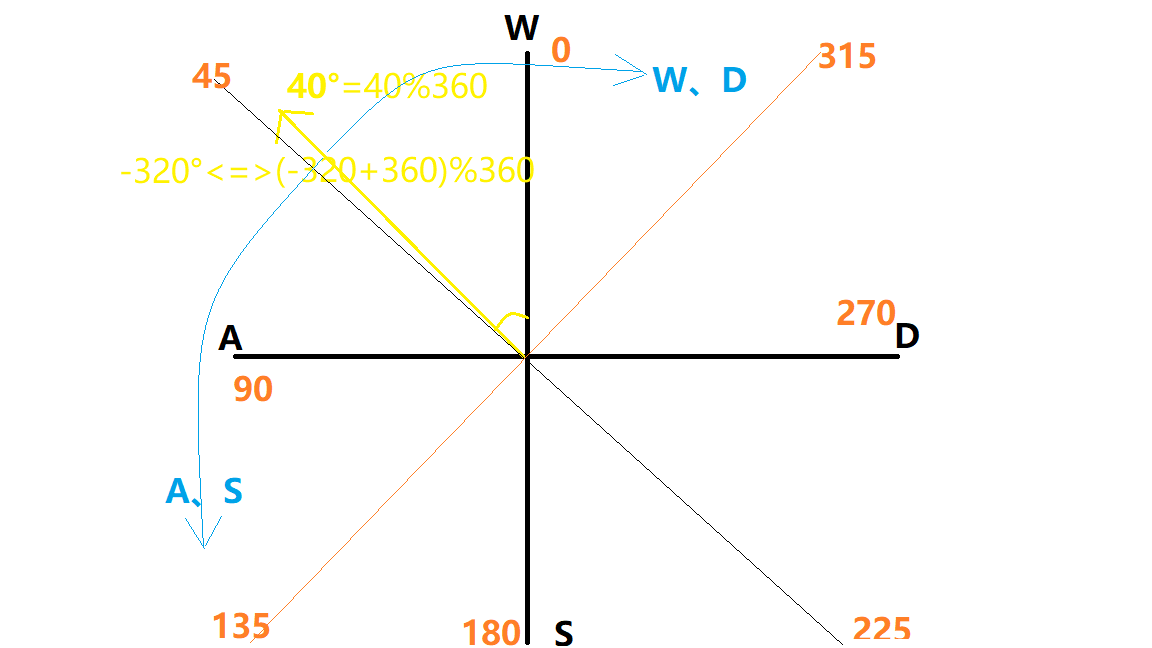
## 任意方向旋转

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++;

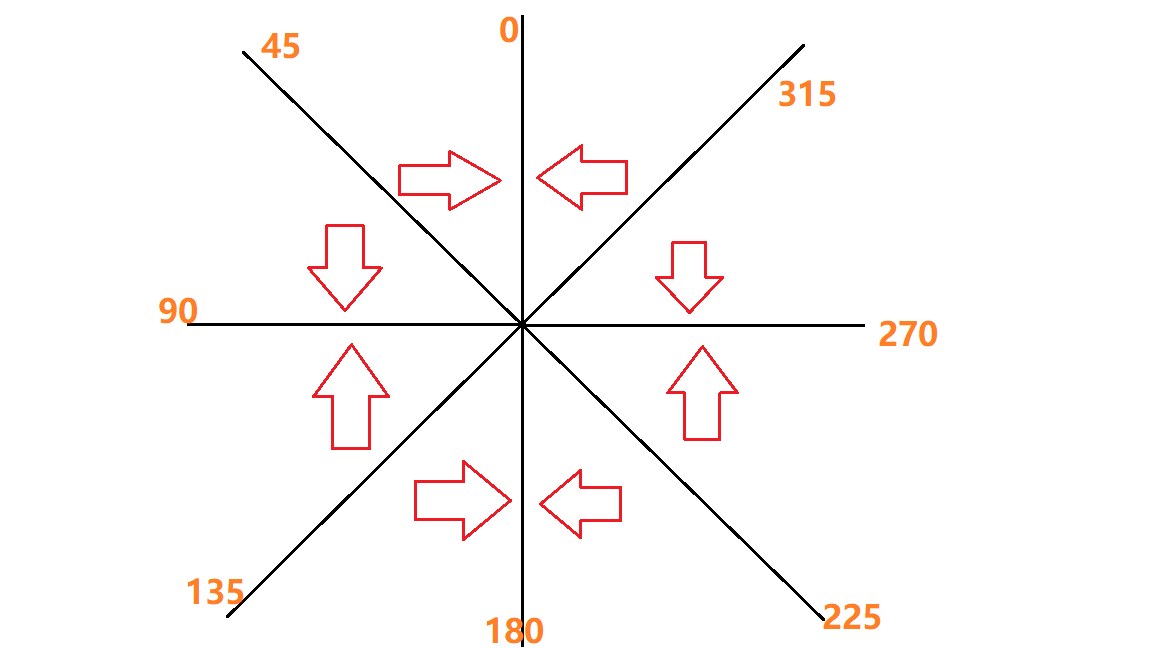
Else Debug.LogError(“...”)

### 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：**

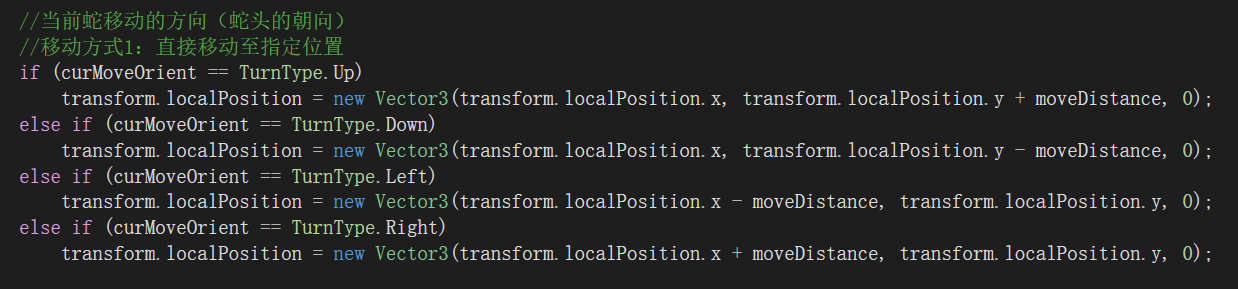
向右移动+向下移动所用的时间 < 蛇的移动频率 ，还没来得及执行上一个移动指令，又有了第二条指令，越过了上一个移动指令直接执行了第二天指令。

**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。

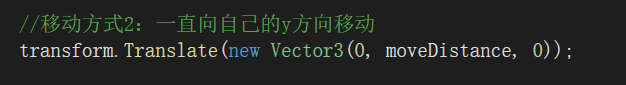
## 蛇头移动

### 移动方式1：直接移动至指定位置



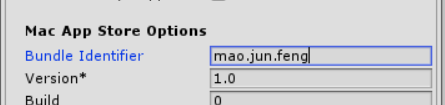
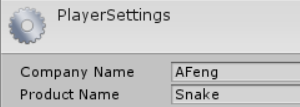
### 移动方式2：一直向自身Y轴方向移动

**注意：这种方式一定要配合游戏物体旋转使用**

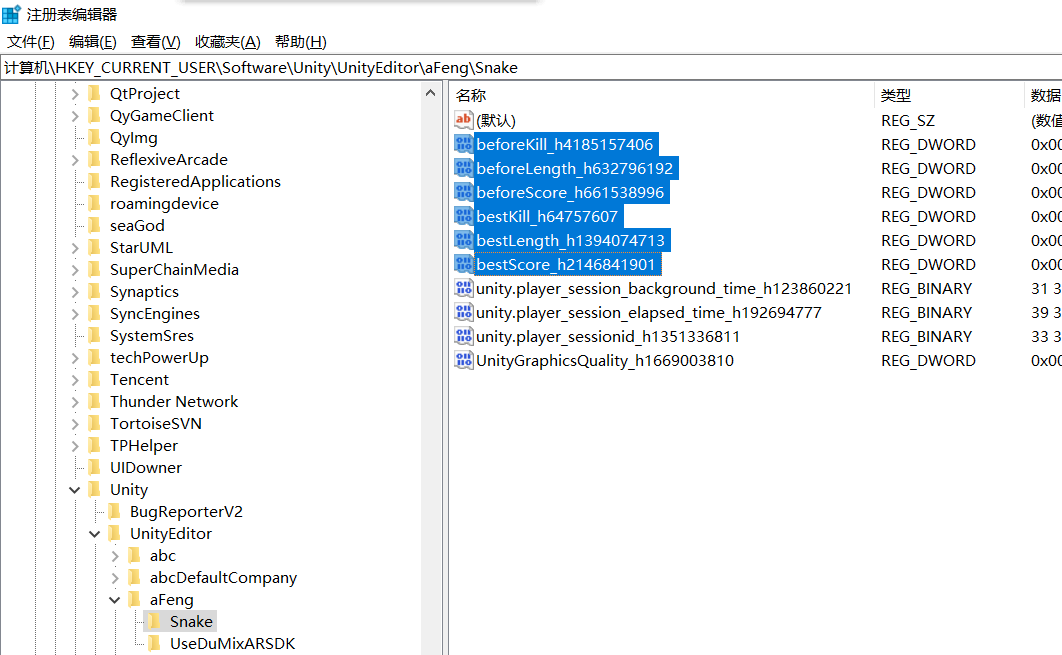


# 6.存档注册表

成绩的注册表：



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



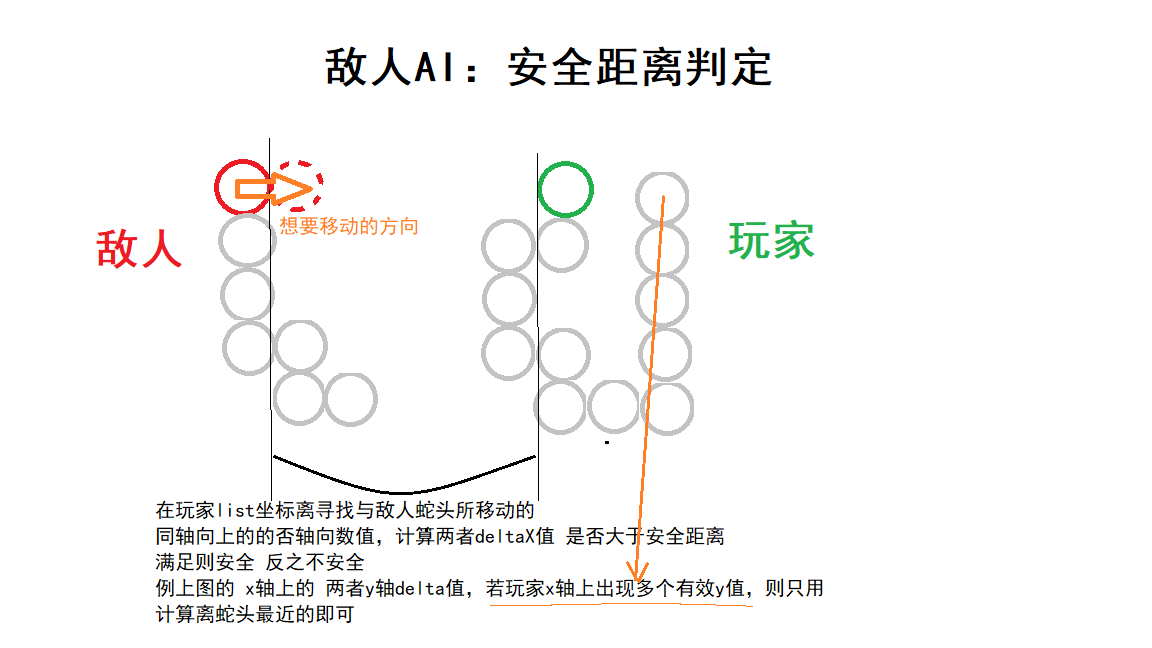
# 7.敌人AI

## 随机自由旋转移动 规则：不允许往当前方向的后方旋转移动

## 避障（可以旋转的方向）

### 躲避玩家：与玩家保持安全距离

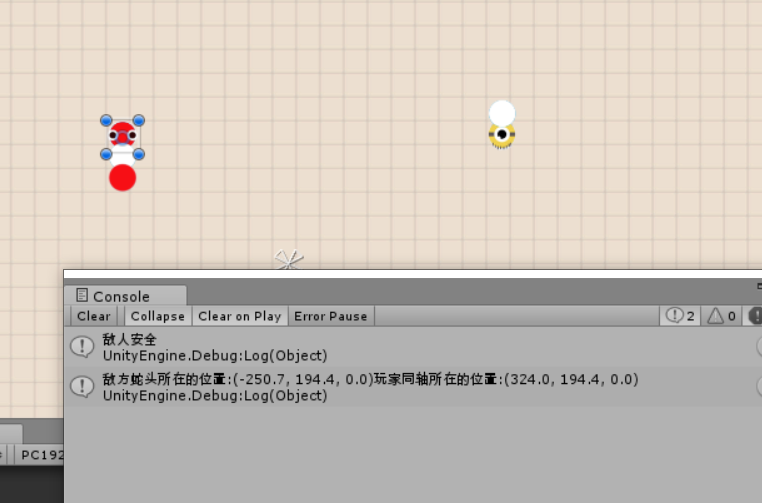
原理： 想要旋转移动后的轴向距离玩家(snakePosList字典坐标)距离 >= 安全距离

原理图: 

实现：离蛇头最近的判定过于繁琐被略去

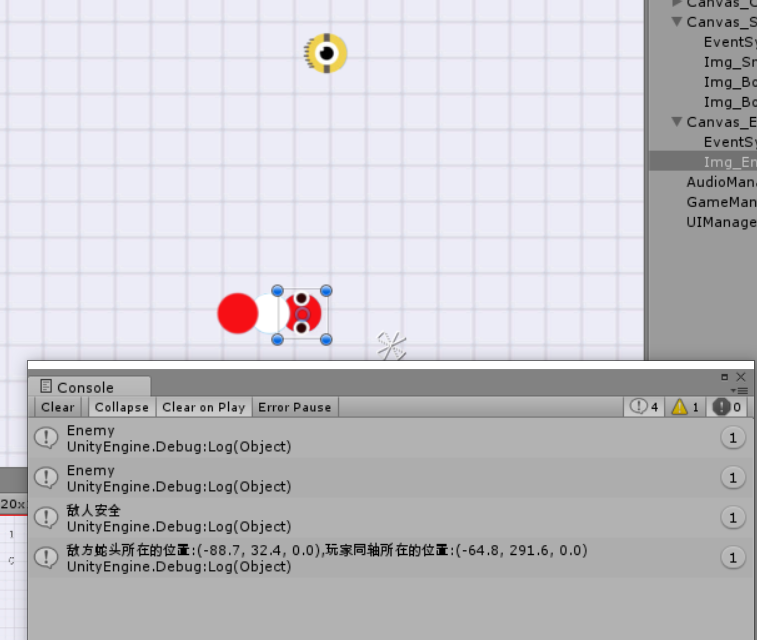
测试 安全距离为200

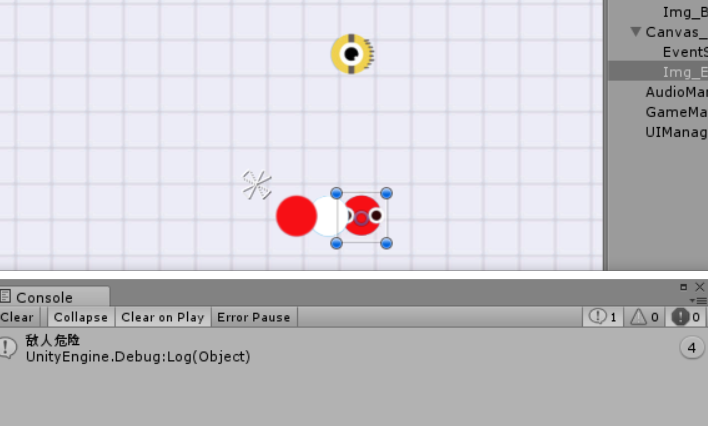
* x轴方向测试





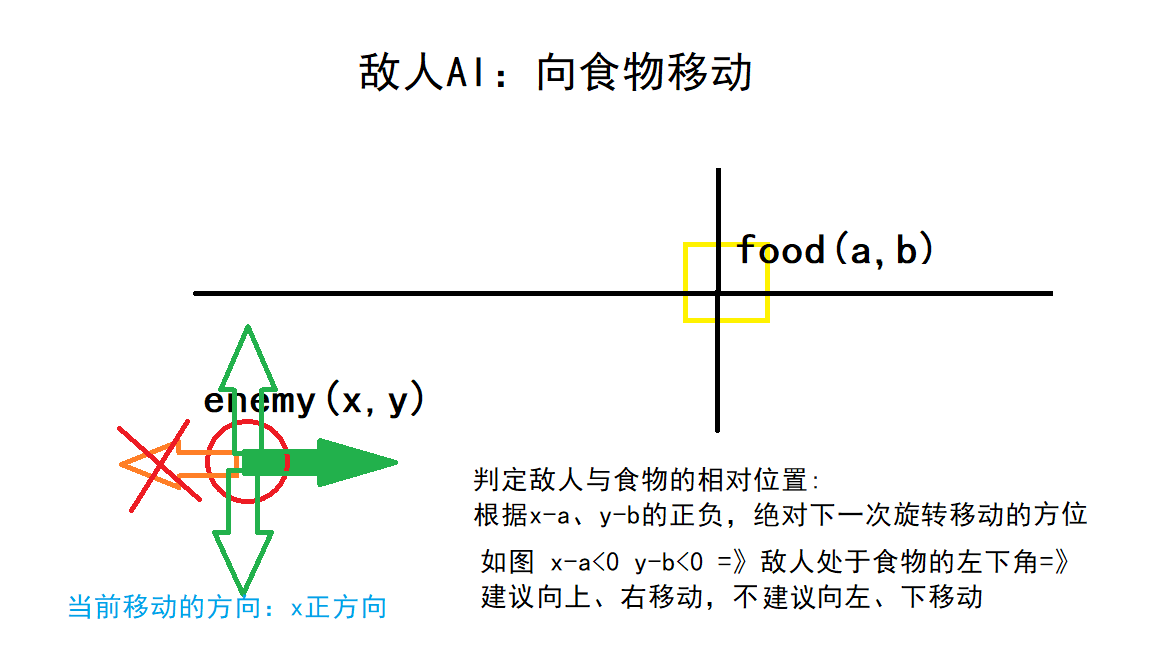
* y轴方向测试：





### 躲避边界

## 3.向食物方向移动（建议旋转的方向）



# 8.制作小地图

使用了HUD Navigation System v2.0.1插件

## 声明：

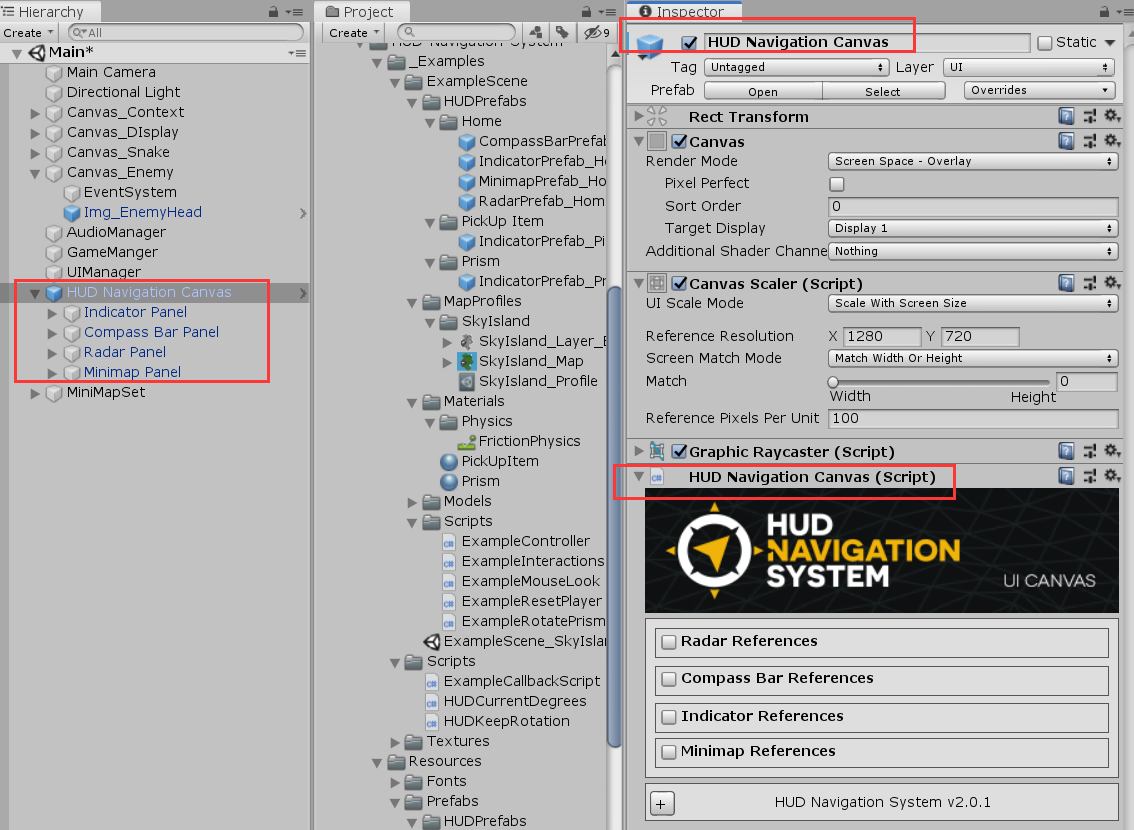
**因插件不适用与5.6版本 ，项目unity版本改为2019.2.13f**

参考样例场景得出以下小结（样例场景为3D转小地图，本项目为2D转小地图，小地图入手视角不同，因为修改了一些脚本逻辑）

## 官方SDK：<https://sickscoregames.gitbook.io/hud-navigation-system/>

## 配置方法：

### 拖入小地图UICanvas预设



核心脚本方法 -- 设置标记的位置

### 配置地图贴图插件

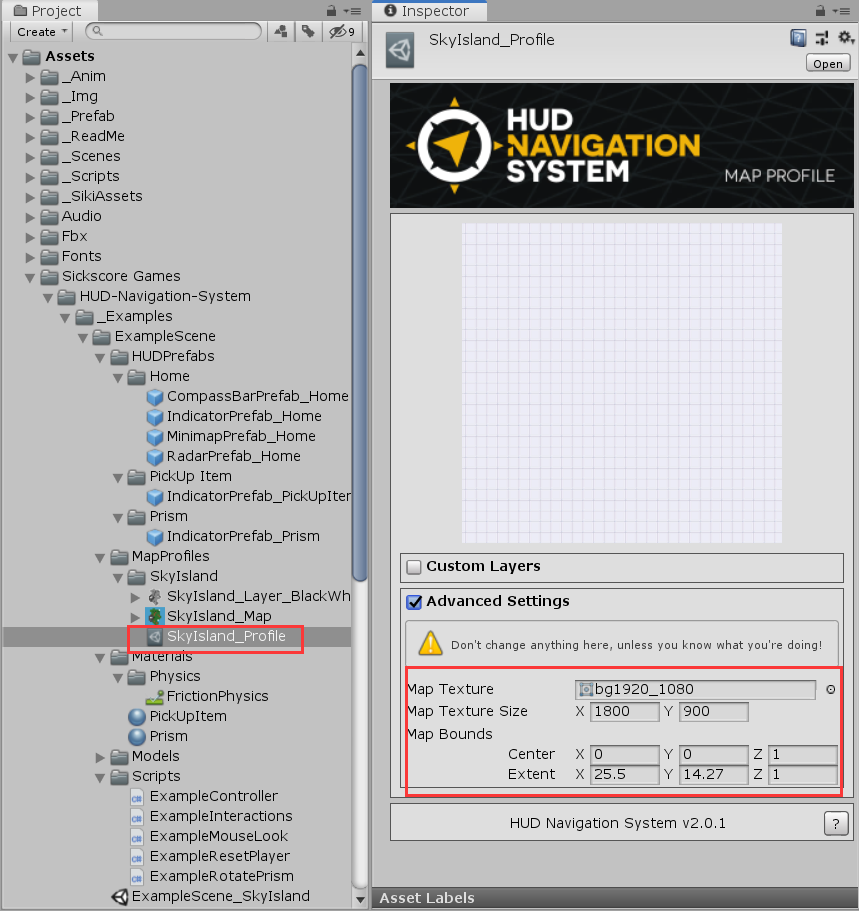
属性：

地图图片

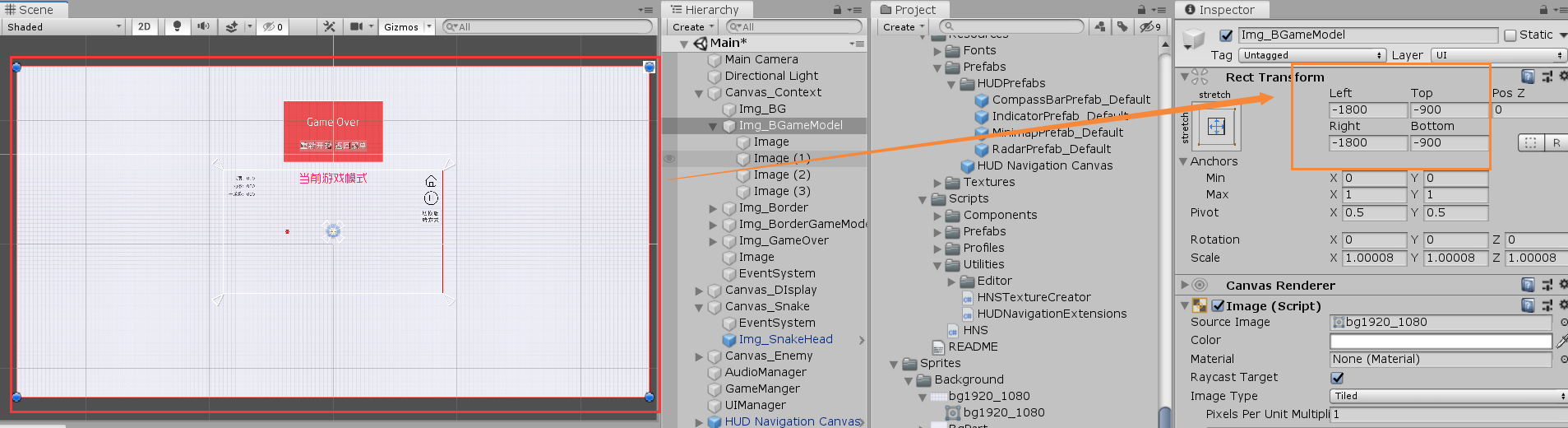
地图图片的分辨率 注意 不是图片实际的分辨率 而是填Transfrom处的分辨率

地图中心位置

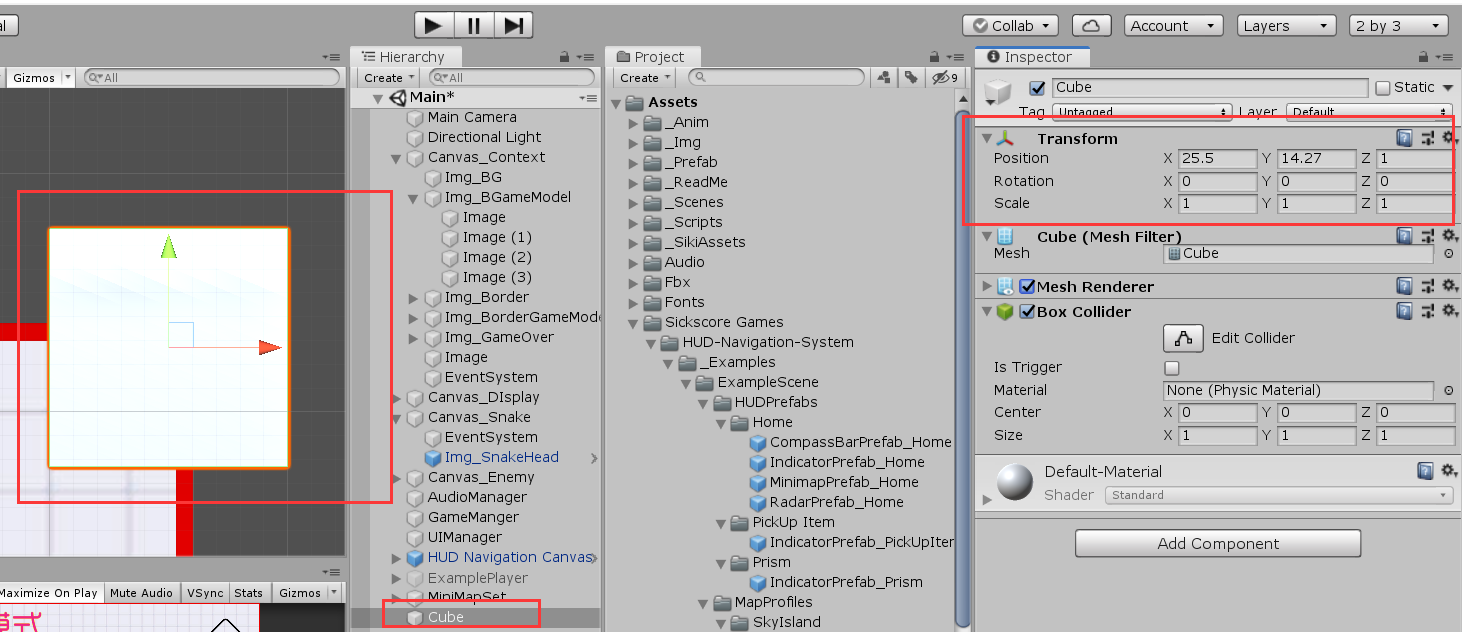
地图边界（右上角）



地图图片的分辨率配置方法：



地图边界（右上角）配置方法

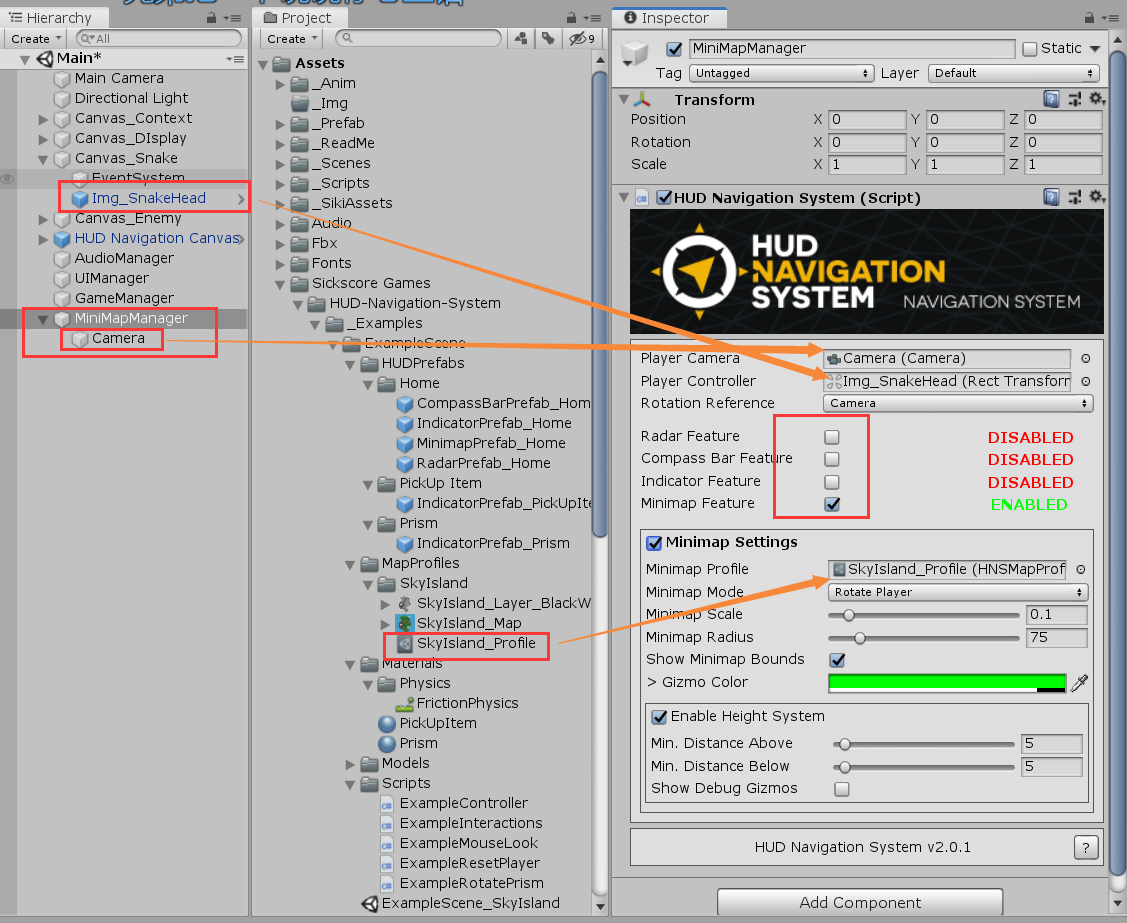


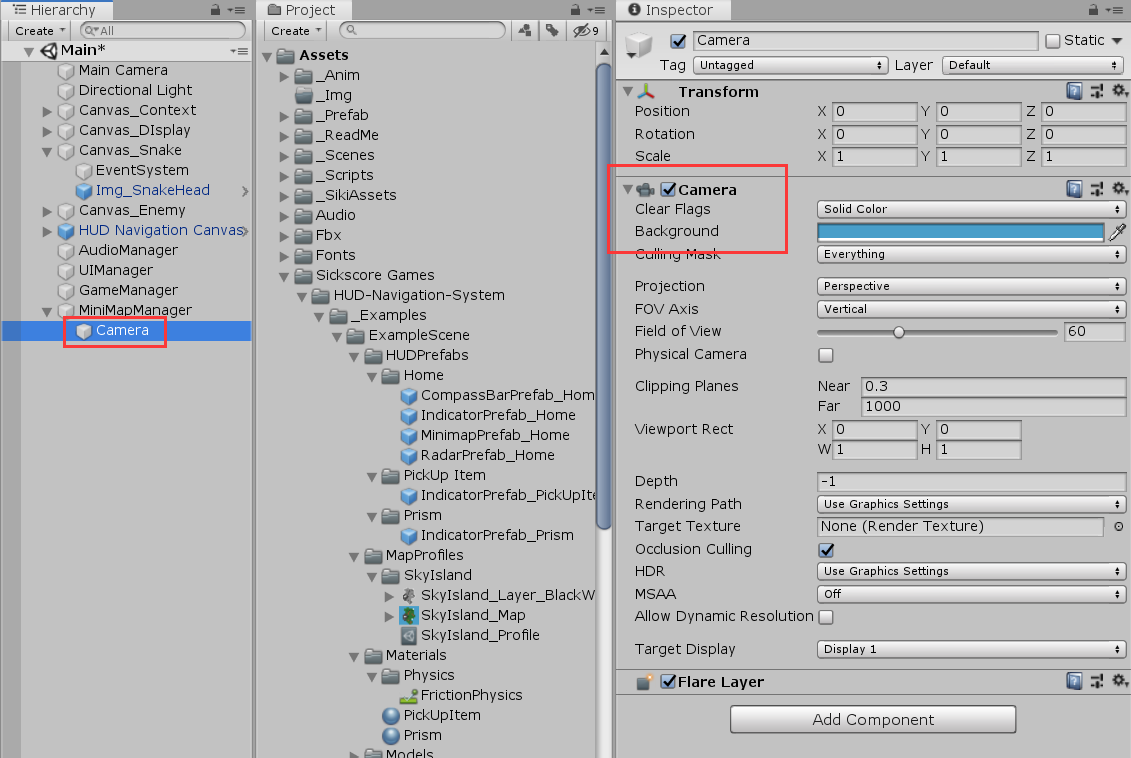
### 配置导航系统 可激活所需要的导航功能

配置方法：创建空对象（或者在玩家对象上）**绑定脚本**，并**添加相机**作为子对象

把小地图相机 和 玩家对象 拖入导航系统

把配好的小地图拖入导航系统

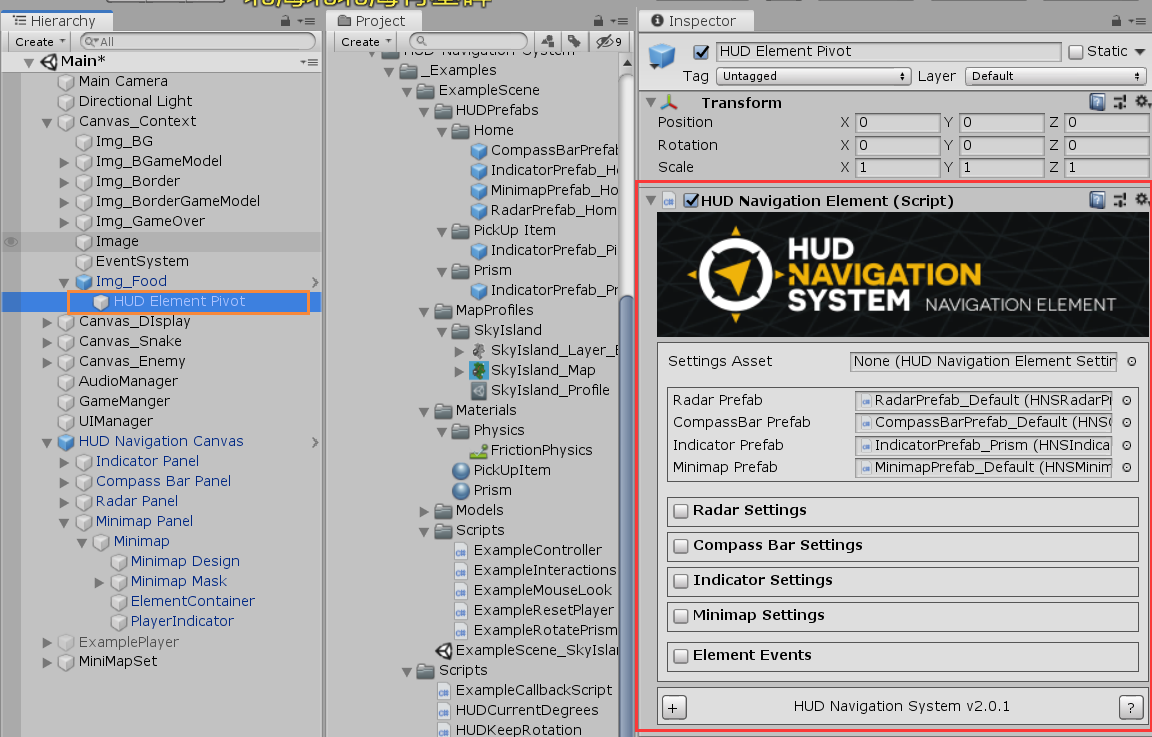




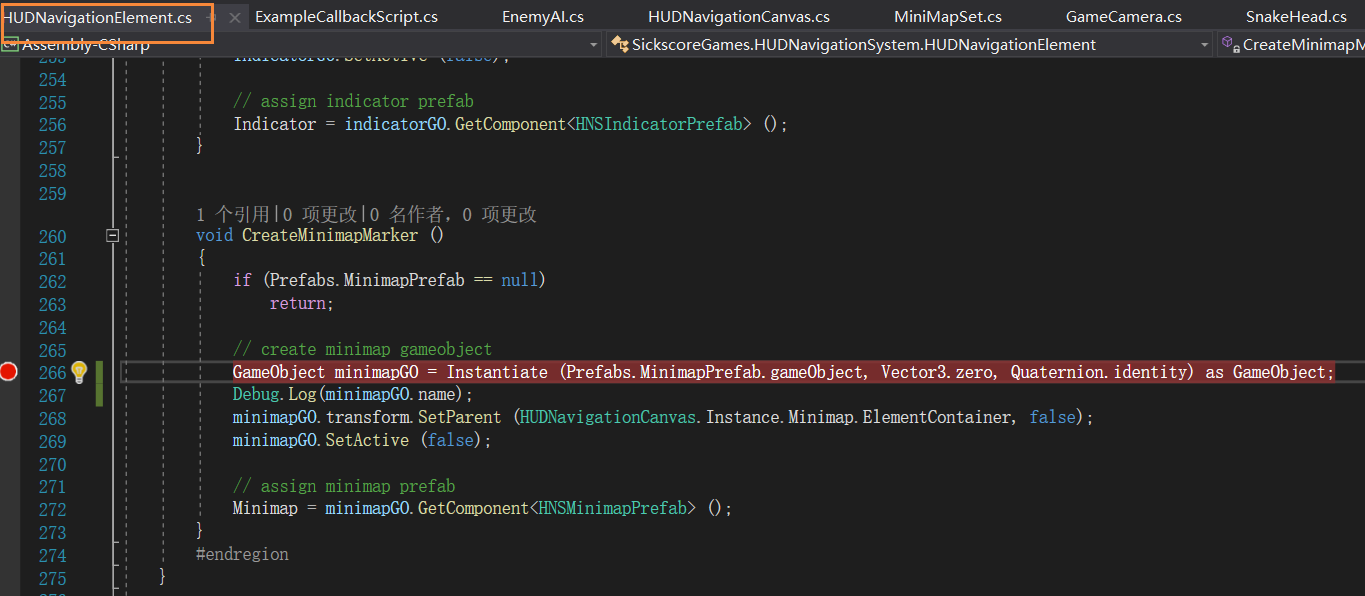
### 添加小地图里其他元素(食物)：

配置方法：在元素（食物）子物体下绑定如下预设（预设可在样例场景中拷贝）

之后可自动实例化该元素的缩略图在小地图资源中

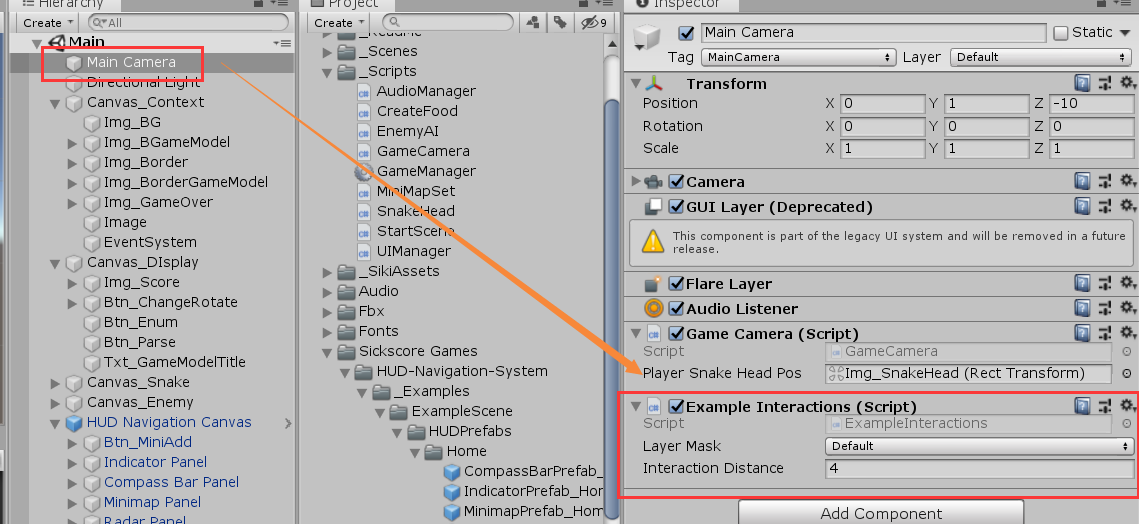


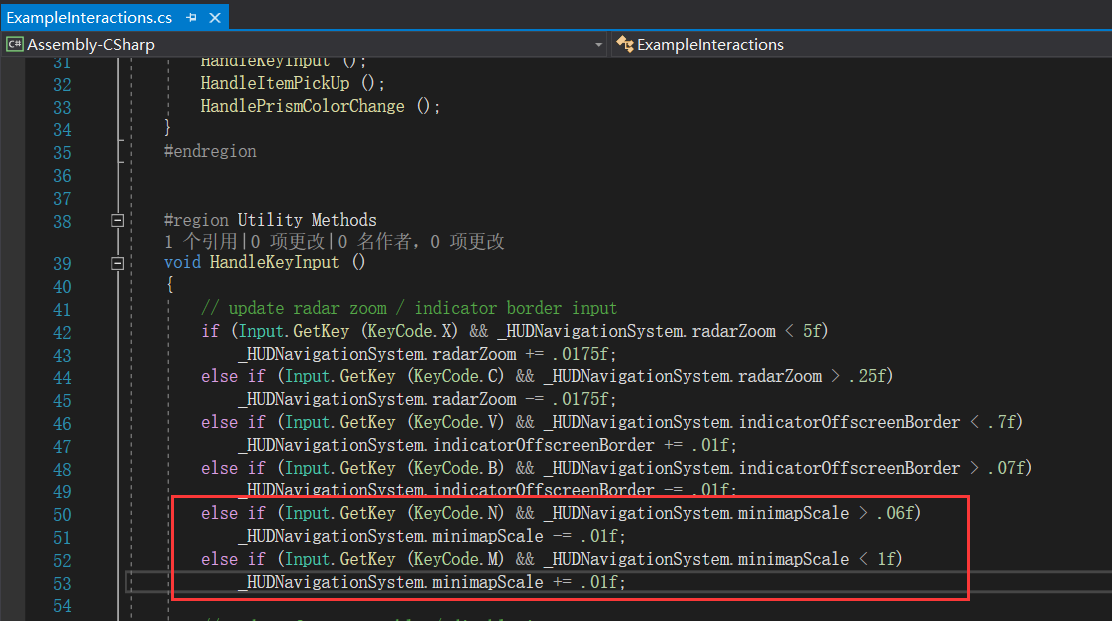
实例化小地图元素



完善：交互 快捷键m、 n 小地图扩大缩小

可放置主摄像机下





### 核心脚本修改 3D->小地图 改为 2D->小地图

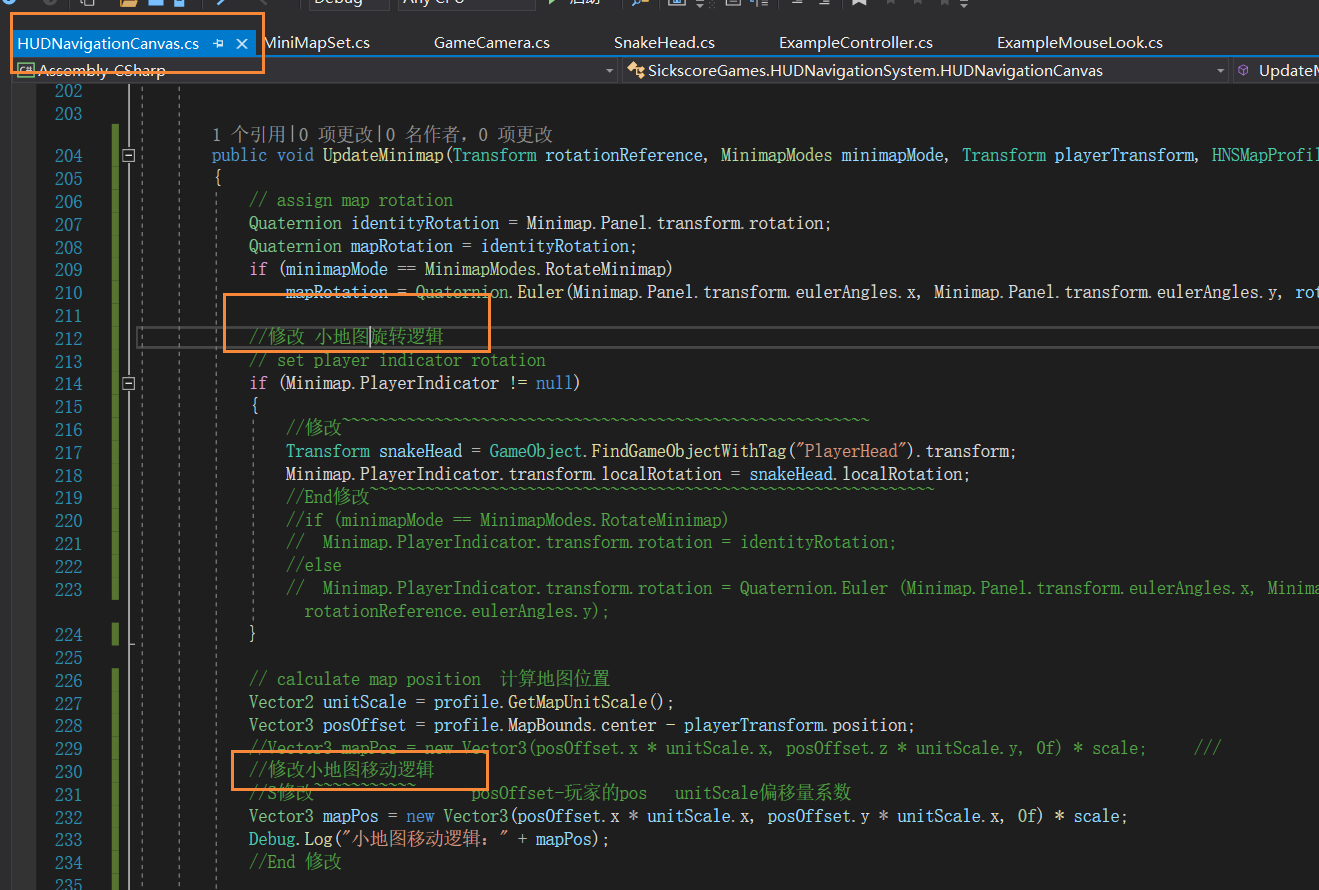
难点：把原工程的3D映射到小地图 改为2D映射到小地图

在3D中 映射到小地图视角为上帝视角（Y轴方向）

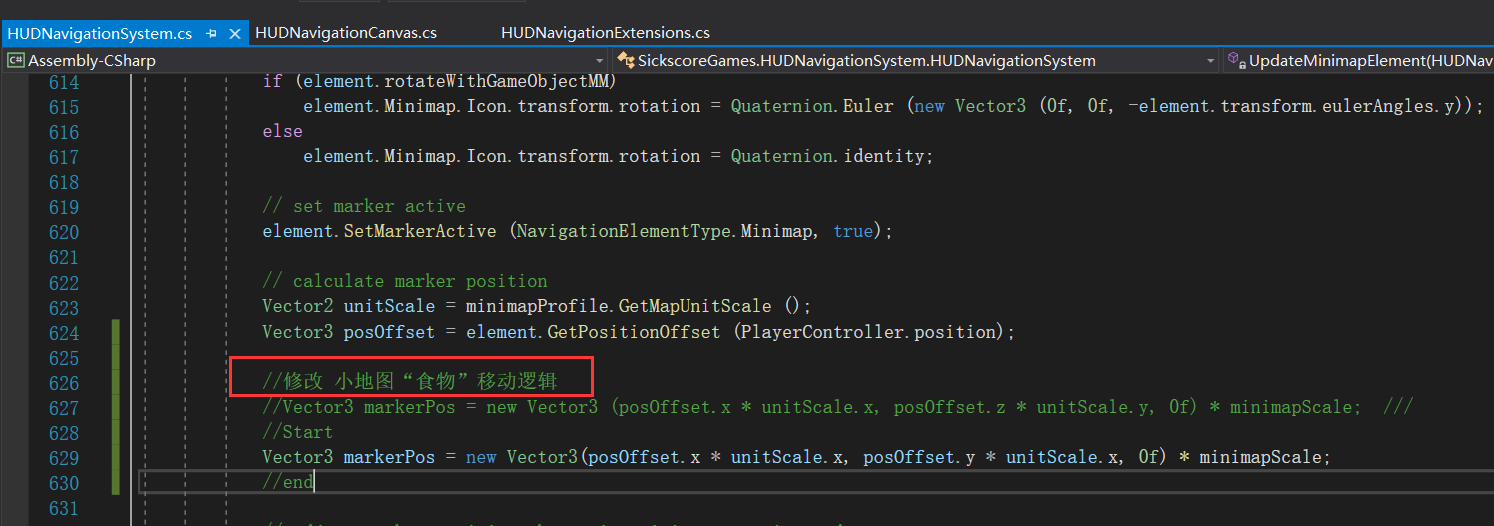
在2D中 映射到小地图视角为Z轴方向

改变了轴向 需要修改旋转、以及移动逻辑

//修改玩家逻辑

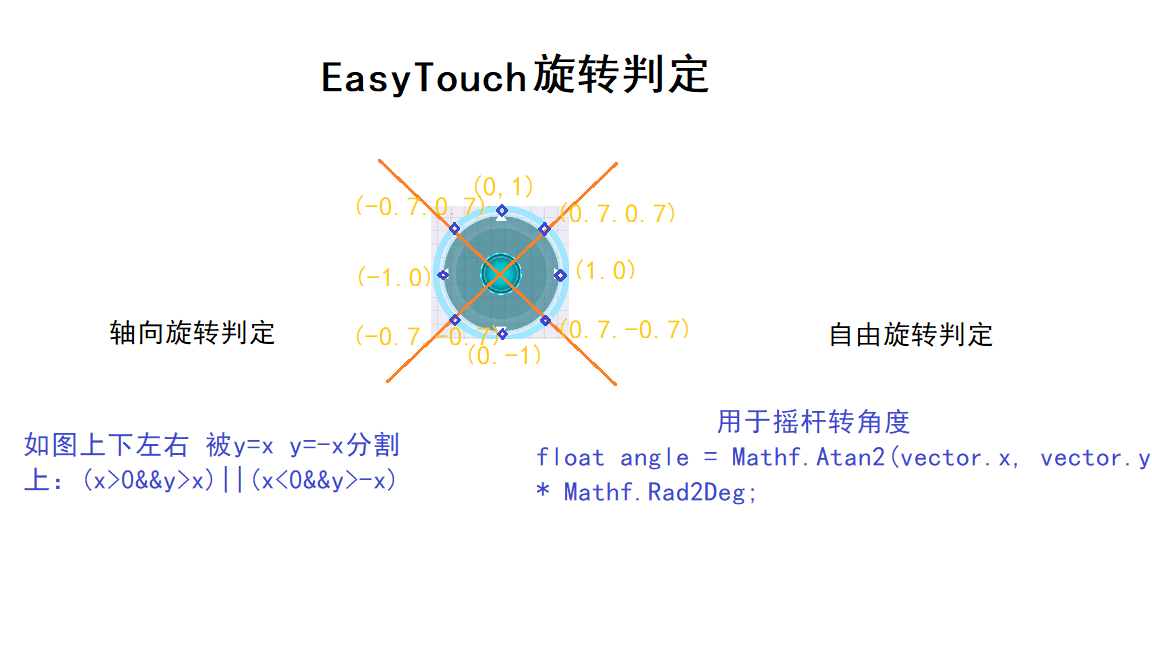


修改小地图“食物”逻辑



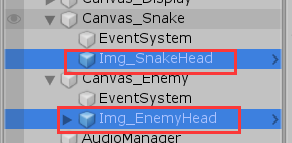
# 借助EasyTouch移动

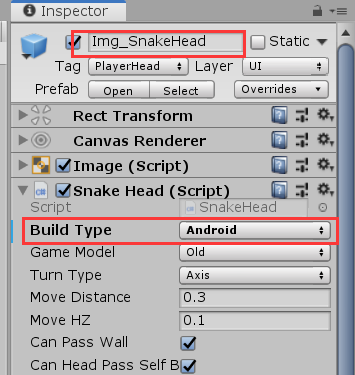
## 旋转原理判定与实现

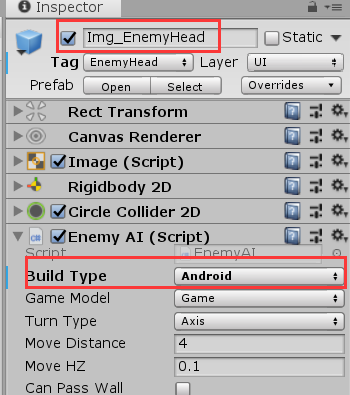


# 10.切换平台方式

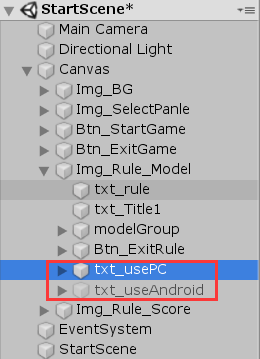
**切换Android和PC时候 以下两个脚本属性要保持一致**







Start场景中的规则面板显示 选其一



# 完结时间：2020年5月6日 19:31:32