NPC行为

参考项目链接： <https://github.com/baponkar/zombie-ai>

## 个人对行为树的理解

行为树给我的感觉就是一个众多if…else的集合，我们要做的就是合理的去定义这些if…else，在什么状态下去判断什么，执行什么，通过众多的条件，根据不同的条件去执行不同的事情，而撑起整个NPC的行为。

## 参考项目结构

定义多个状态，每个状态单独一个类，继承同一个接口，接口中实现了三个方法，分别在状态开始时执行，处于该状态时一直执行和退出时该状态时执行。

文本

描述已自动生成

通过一个状态管理类管理所有状态类，根据不同的事件向该类传递不同的状态类型进行切换状态，并在不同的时机执行不同的方法。

文本

描述已自动生成

文本

描述已自动生成

RegisterState方法用于注册一个新的状态，它将传入的NPCState对象添加到states数组中的对应位置。

GetState方法根据传入的状态ID返回对应的NPCState对象。

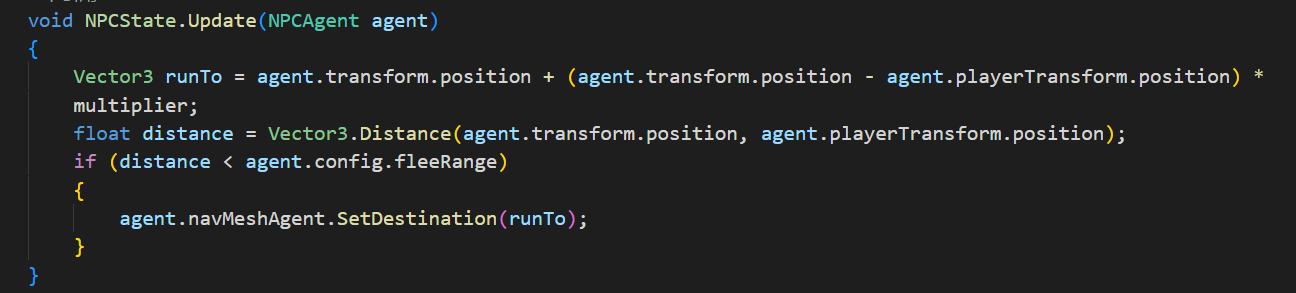
Update方法用于更新当前状态，它调用当前状态的Update方法，并将NPCAgent作为参数传递给它，并在MonoBehaviour中的Update中被调用。

ChangeState方法用于改变当前状态，它首先调用当前状态的Exit方法，然后将新状态的ID赋给currentState变量，最后调用新状态的Enter方法。

整体的执行方法是：每帧执行当前状态的Update方法，在切换状态时，先执行切换之前状态的Exit方法即退出时执行。再执行切换后状态的Enter方法即开始时执行。

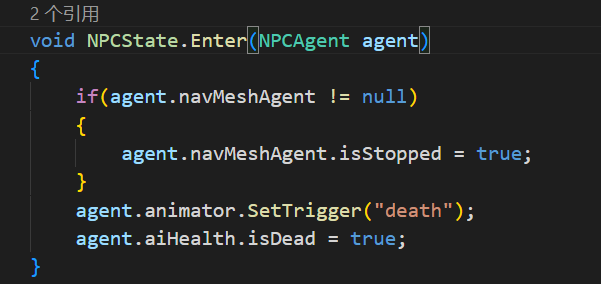
这里的行为不能并行执行，需要并行执行的话，该项目使用的方法是采用多脚本

## 逃跑状态



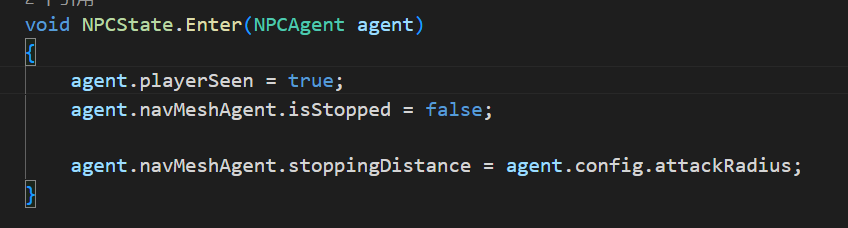
用自己的位置减敌人的位置拿到一个向量，自身位置加上该向量乘一个系数，达到向反方向跑的目的，直到跑出一个距离。

## **没血时死亡**



关闭寻路，播放死亡动画。

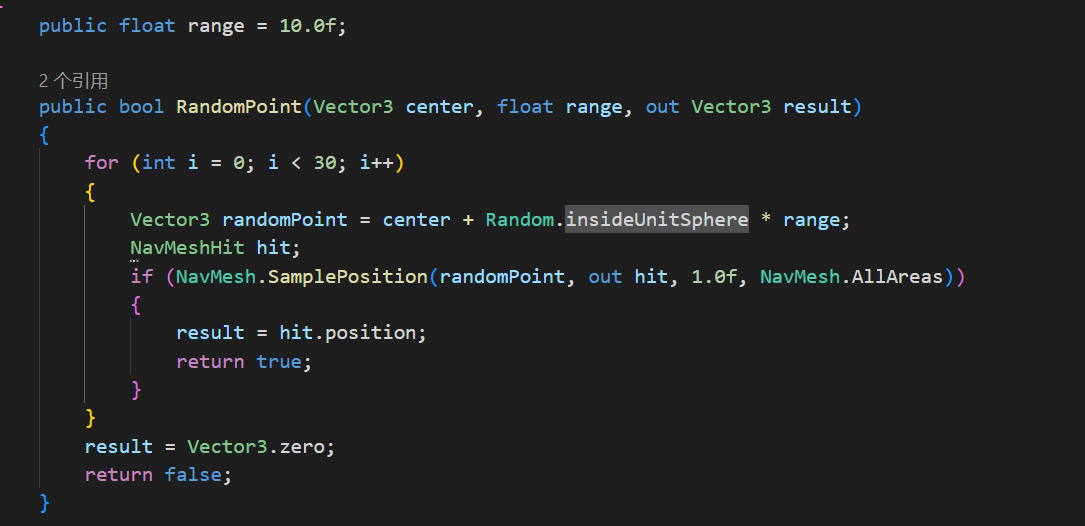
## 发现敌人时追击



追击就是将玩家位置给寻路组件，实现追击。

## 巡逻

巡逻时首先不断的随机位置移动，随机移动的方式是通过Random.insideUnitSphere随机生成一个向量乘系数以后得到一个向量，用自身位置加该向量，即可得到目标位置，再判断目标是否在地图上，在地图上就移动，不在继续循环，若30次都未判断成功，则跳过该次执行，等待下次执行。



并根据自身血量进行不同的判断，血量少时发现敌人逃跑，血量多时发现敌人追击。发现敌人的方法是通过射线判断前方扇形位置是否有玩家。

## 靠近敌人时攻击

在靠近敌人后播放攻击动画。



同时在Update中判断血量，根据血量做出调整。