实现一个吃豆人(Pac-Man)世界的案例：

1 具体地描述world的状态

世界的空间属性：

该世界应当有一个地图属性 规定可通过和不可通过的地块

该世界的所有实体都应该在整数格的位置上

世界的实体属性：

该世界至少应该有三种实体 ：

吃豆人：

属性：位置

方法：移动

解释：

吃豆人可以通过移动的方法吃豆子而不用专门设置吃豆子的方法 只要吃豆人和豆子在同一位置 就判定吃豆人吃到了豆子

鬼魂：

属性：位置

方法：移动

解释：

鬼魂和吃豆人逻辑类似 只是对鬼魂而言 吃豆人才是他们的豆子（他们吃吃豆人 就像吃豆人吃豆子一样） 唯一的不同在于吃豆人会动

豆子：

属性：位置

方法：无

解释：

豆子不会移动 也不需要方法 只需要在一个位置上等待被吃豆人吃

一些属性和实体直接相互作用的逻辑：

吃豆人和鬼魂不能移动到不可通过的格子上

吃豆人和豆子在同一格子时 判定为吃豆人吃到这个格子的豆子

鬼魂和吃豆人在同一个格子上时 判定为鬼魂吃到吃豆人

鬼魂和吃豆人不能跑到地图外面去

属性的变化：

吃豆人每回合读取AI的指令 移动一次

鬼魂依照特定算法 每回合移动一次（也可以多使用multi agent的方法 让AI决定他们的行为）

世界的生成：可以制作一个内置的默认的地图 初始化世界

地图中只应该有一个吃豆人

所有实体不能出现在不可通过的格子上

吃豆人和豆子不能初始化在同一个格子上

一个格子最多只能有一个豆子

2 构思如何实现这些属性 并实现之

实现过程：

创建一个新的world\_project 并命名为pac\_man\_world

1 我们首先在world\_project文件夹下 复制一份blank\_world 然后将其改名为pac\_man\_world

2 将文件夹下的所有文件和文件名中的”blank”和“Blank”改为“pac\_man”和” “Pac\_man”（可以直接用文本替换功能实现这一点）

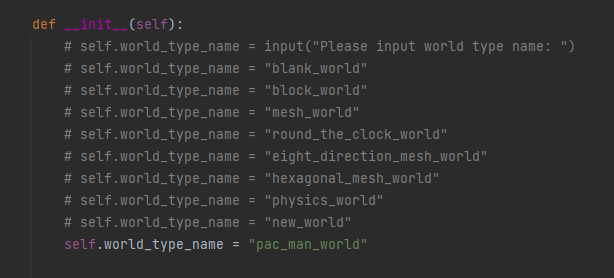
需要注意的是新文件夹下的entity文件夹中的blank\_entities.py也要改名为pac\_man\_entities.py

3 为了验证修改没有出问题 最好跑一遍测试一遍

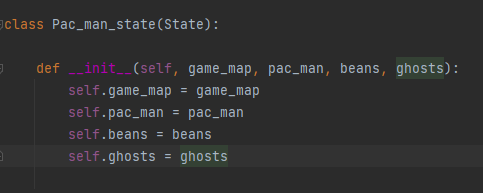
在world\_env\_interface.py的\_\_init\_\_中定义self.world\_type\_name = "pac\_man\_world"让MTEAC类指向pac\_man\_world

然后运行main.py(注意设定ai\_mode=1)

如果运行没有报错 则说明创建成功了



在state.py中定义属性

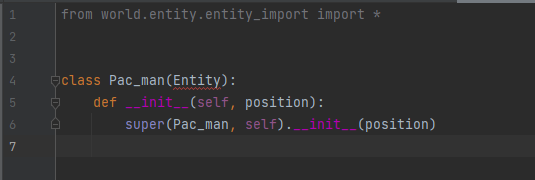


定义实体

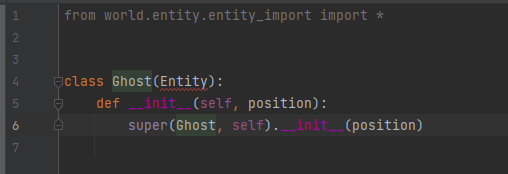
首先在pac\_man\_world中的entity文件夹下创建相应的文件



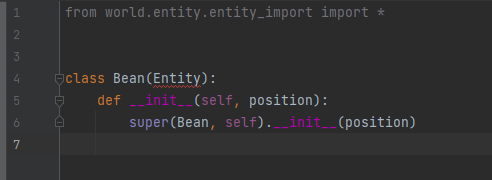
定义Pac\_man



定义Ghost



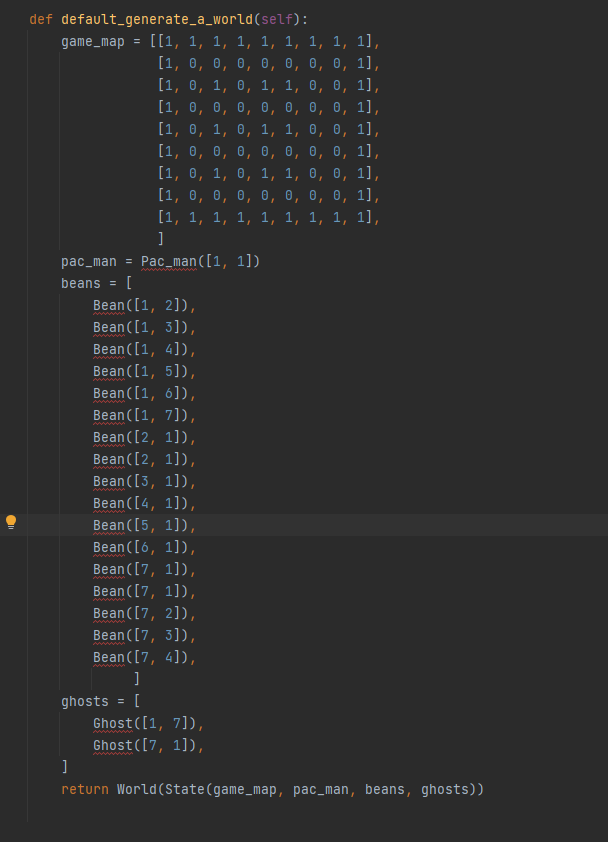
定义Bean



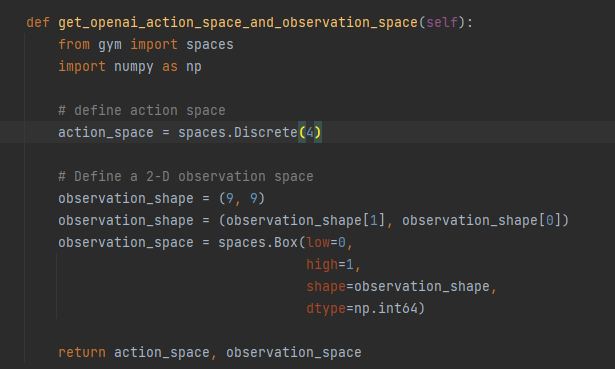
注意注意：定义entity的文件的文件名和其中的类名需要一致 类名的首字母需要大写

如果不遵循这个规则可能造成不能使用默认读档器

在generator中定义这些属性如何被初始化

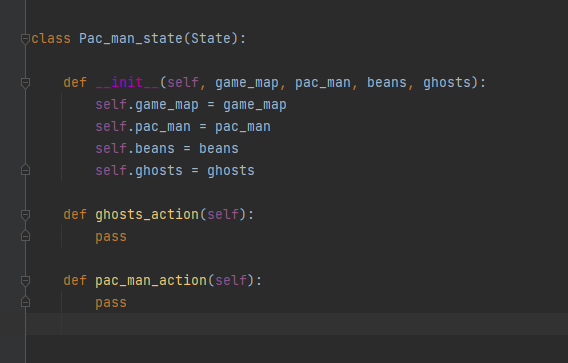


定义状态空间和行动空间



定义state的运作逻辑

首先思考一回合中让属性变化的方法有哪些



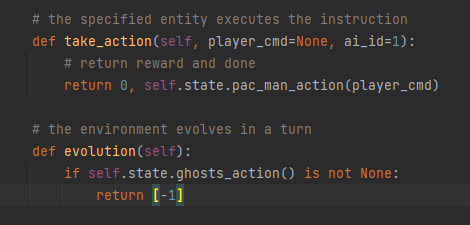
然后在world的take\_action()和evolution()中分别调用对应的方法

take\_action()是读取ai指令 并让指定实体进行操作

take\_action()会返回reward和done给openAI

evolution()是除了ai操作的实体以外的实体的行为以及其它属性的变化

若环境运作使得某个ai操作的entity的游戏结束了 则返回一个数组替代take\_action()中返回的done 否则返回None

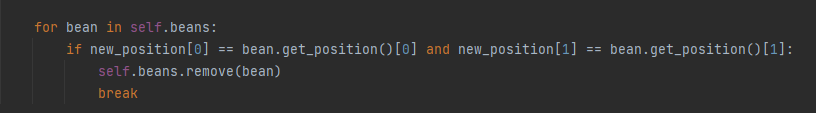


然后具体实现变化的逻辑

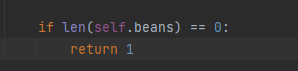
吃豆人通过移动变换位置



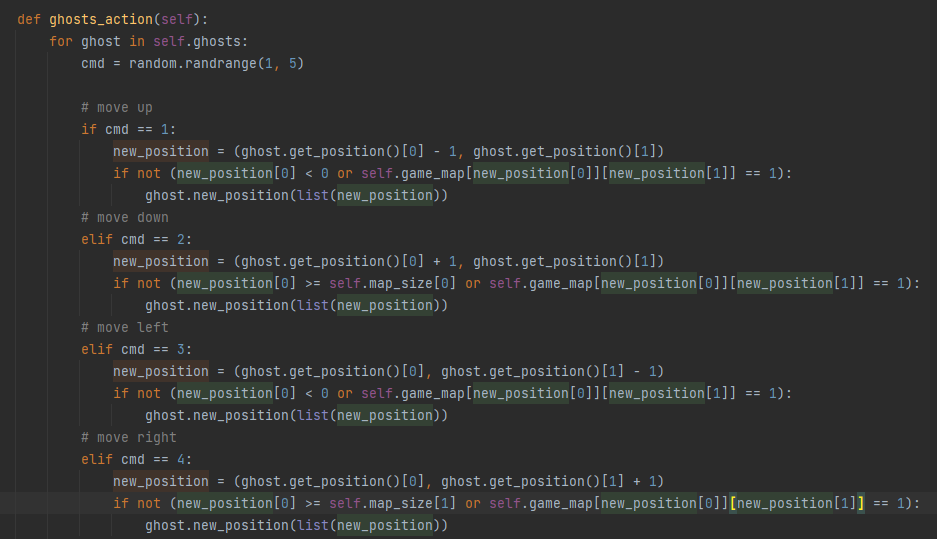
若新位置上有豆子 则豆子被吃掉



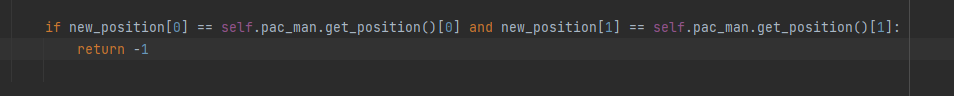
若豆子被吃完 则游戏结束 胜利



鬼魂通过移动追逐吃豆人（这里采用一个随机散步）

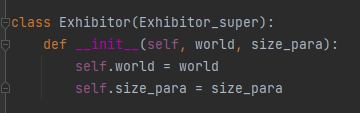


若和吃豆人位于同一位置 则吃豆人被吃掉 吃豆人游戏失败

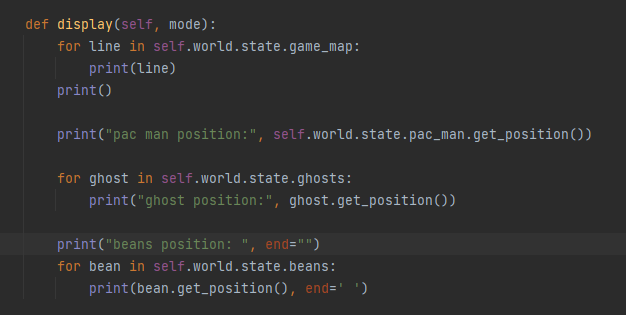


实现简单的可视化

初始化Exhibitor（world是world project的实例）



在display()中实现可视化（display方法相当于每次openAI gym中的render()方法）

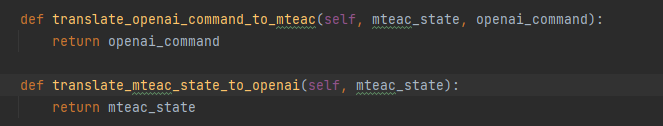


测试

设定成功和失败条件



由于openai的行动指令和world\_Project内的一致 我们不需要做任何二者之间的翻译



运作结果：

