Example of Building a “Pac-Man world”

1 To describe the status of the world

The Attribute of the world.

This world should have a map attribute to define whether a block can be pass through.

All entities in this world should allocate at integer positions.

The attribute of entities of the world

This world should have at least 3 type of entities

Pac-Man:

Attribute: Location

Functions: moving

Explanation:

Pac-mans should able to to collecting beans by moving over them, That is mean if pac-man is at same location with beans, he gets the beans.

Ghosts(enemy):

Attribute:

Location

Functions: moving

Explanation: Same logic with pac-man collecting beans, but for ghosts, pac-man are there beans to collect.

beans:

Attribute:Location

Functions:None

Explanation: beans can not move, so it does not need any functions, they just sit there for pac-man to collect.

Other logic of how to how does entities interact with each other

Pac-man and ghosts can not move to blocks that can not be pass through,

When pac-man is at same position with beans, pac-man collects the beans

When ghosts is at same position with pac-man, they kills pac-man

Change of attributes:

Pac-man read instruction from AI agent and move every round.

Ghosts move every round by their moving algorithm.( or use another AI agent to control it)

Generating the world: we can initialize a default map

Each map must have only one pac-man

No entities should locate on blocks that can not pass through

Pac-man can not be initial on the same position

Every position can have at most on Beans

2,structuring on how to achieve these, and achieve them

How to achieve it:

Create a new “world\_project” and name it “pac\_man\_world”

1, we copy a “ blakc\_world” from “world\_project” and rename it “pac\_man\_world”

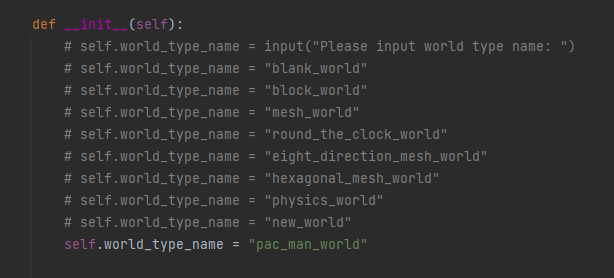
2, change every “blank” and “Blank” to “pac\_man” and “Pac\_man” ( you can do this easily by ctrl + f) PS: in you new file, the “blank\_entities.py” Should be rename as “pac\_man\_entities.py “as well.

3, better run this to make sure nothing goes wrong.

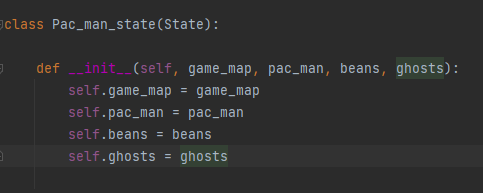
Define “self.world\_type\_name = "pac\_man\_world" in the “\_\_init\_\_” under “world\_env\_interface.py”, and let MTEAC class points to “pac\_man\_world”

Then run “main.py”

(if no error shows up, that means we successfully create the “pac\_man\_world”



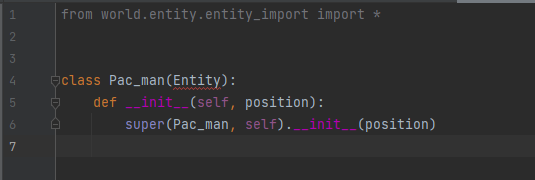
Define attributes in “ state.py”



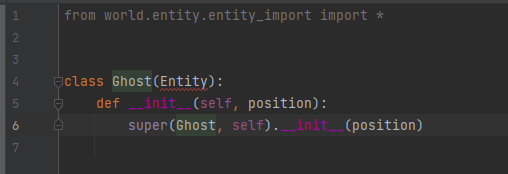
To define entities, first create their file under the “entity” folder in the “ pac\_man\_world”



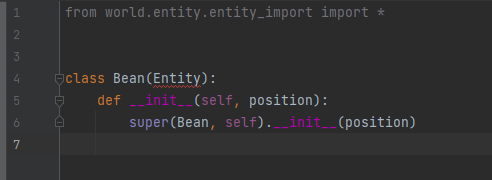
Define pac\_man



Define Ghost



Define Bean

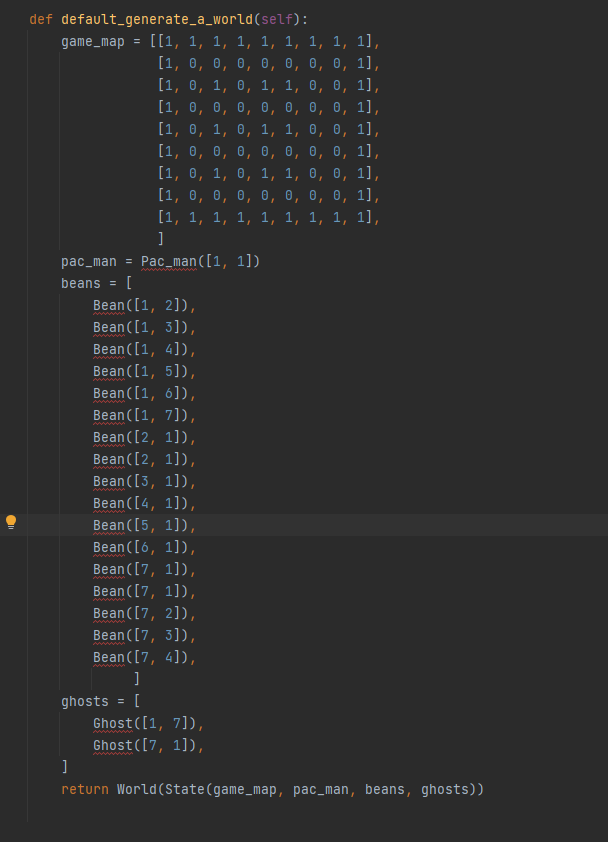


Attention !

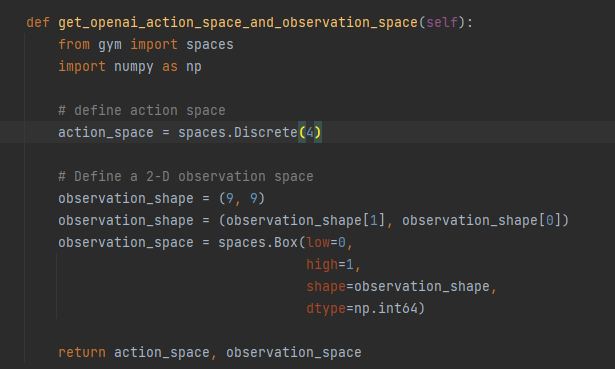
The file used to define entities needs to have same name with the defined entities class, and caps the first letter.

If not, there might be error in the save and load function

Define how these attribute will be initialized in the “generator”

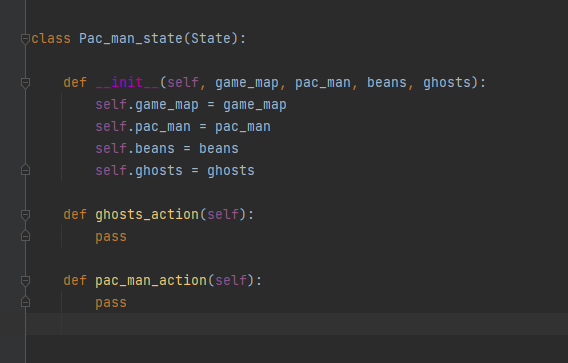


Define actions spaces



Define the logic of state

First we think about what is the functions that can cause changes on attribute in each rounds

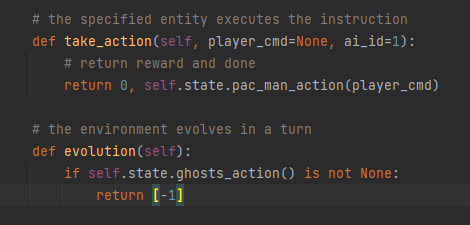


Then we call these functions in “take\_action()” and “evolution()” under”world”

“take\_action()” is to let entities reads instructions from AI and can be control by it, it returns “reward” and “done” to OpenAI

“evolution()” is other changes aside from the entities that controls by AI.

If some AI controlled entities cause a game over, returns an array to replace the “ done” in “take\_action()” , if not, return “None”

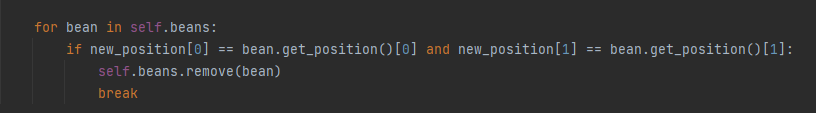


Then define logic of changes:

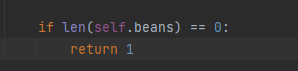
Pac-man changes positions by moving



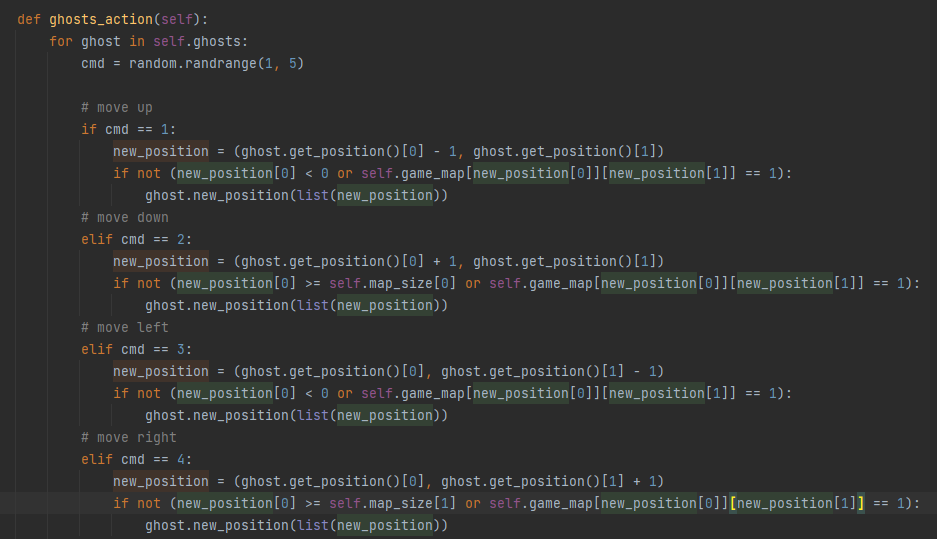
If there are beans on new positions, pan-man collect them



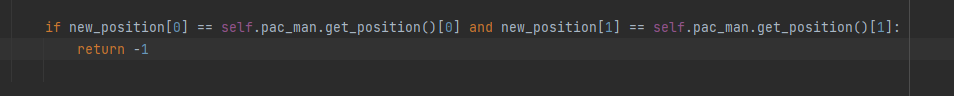
Game wins when all beans are collected



Ghosts chasing pac-man by moving

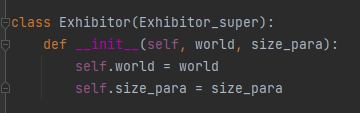


If ghost is at same position, then pac-man dies and game over

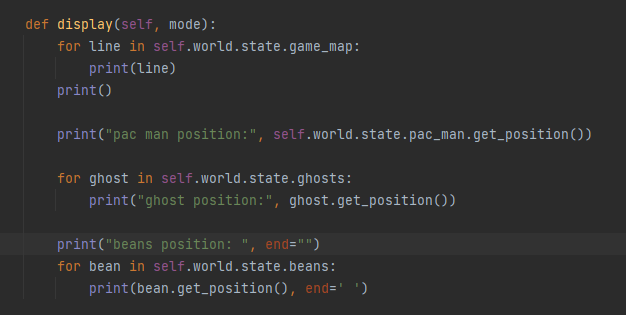


Sample visualization

Initializing Exhibitor



Achieve Sample visualization in “display()” ( display() function is equivalent to “render()” function in Open AI gym)

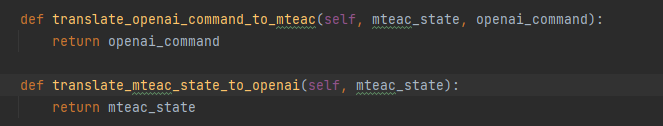


Test

Setting win conditions and lose conditions



We do not need translate our instruction, because action instruction in “world\_project” is the same as in Open AI.



result

