**How to use block\_world**

“block\_world” is an simple example of “world project”, building this is for demonstration of how to build a simple “world project”

**Description of “block world”**

Spatial attributes:

This world should have a map attribute to define whether a block can be pass through or not

All entities in this world should be place on integer value position

Should use for directions to judge if blocks are next to each other

Entities attribute in block world:

This world should have at least two type of entities:

Human:

Attribute: position

Functions: moving

Explain: human can push boxes by moving, if the a box is in the way of human, the box moves with human

Boxes:

Attribute: position

Functions: moving\_to\_new\_position

Explain: boxes does not move themselves, but they can bu pushed around by human

**Logic of how entities interact with each other:**

Human and boxes can not go outside of map

Human and boxes can not go through blocks that can not be pass through

**Changes in attribute:**

Each round, human reads instructions from AI agent and move, changing position of human and possibly position of boxes

**Generation of the world:**

There should be only one human on the map

There can be multiple boxes on the map

Human and boxes should be inside the map

Human and boxes should not be at the block that can not be pass through

Human and boxes should not be at same position

**Use blank\_world to create new world project**

1, Name your new world, eg: new\_world, (PS: name has to end with “\_world”)

2, Copy “ blank\_world” from”world\_project” , change the name to new\_world.

3, change “black” and “Blank” to “block” and “Block” in the name of all files under “new\_world” folder .

1. To ensure changing name does not cause any error.

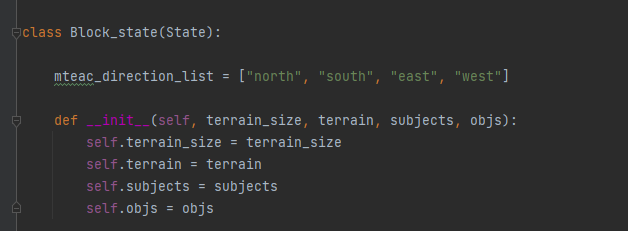
Define “self.world\_type\_name = "block\_world"" under “world\_env\_interface.py” to make MTEAC class points to “new\_world”. then run”main.py”( remember to set ai\_mode=1)

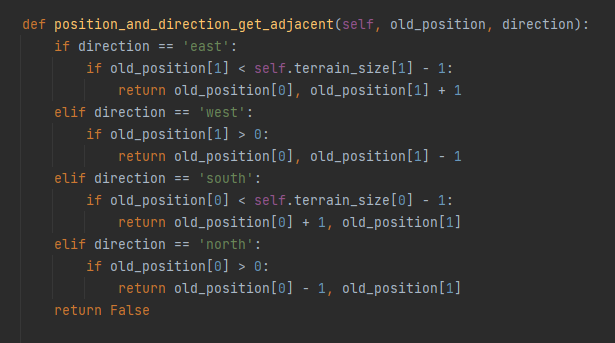
If there are no error occur, the new world is create successfully.

Define of status attribute

Define of Spatial attribute

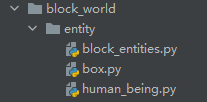
Define them in “state.py”





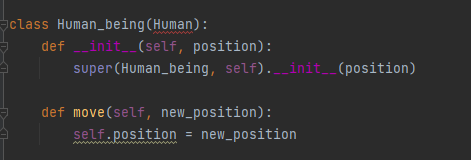
Define of entities attribute

Creating human\_being.py and box.py

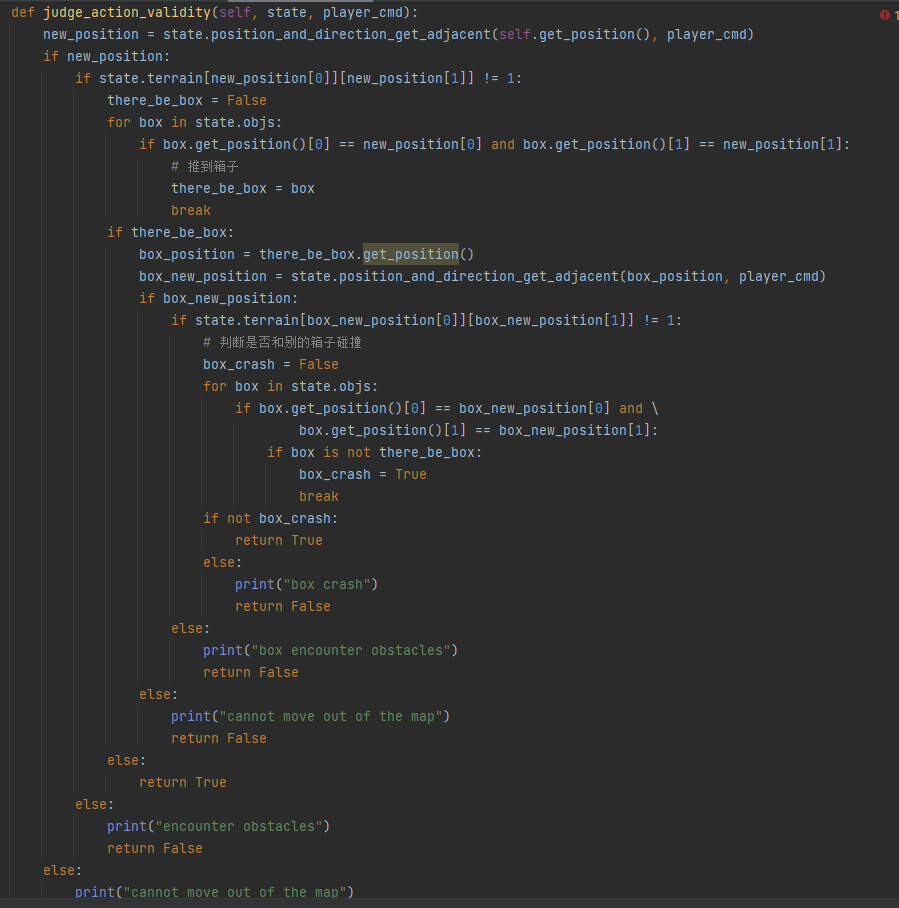


Define human in “human\_being.py”

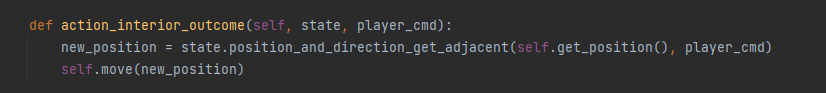
Function of moving



Define conditions of moving, also a legitimacy judgment



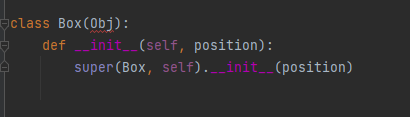
Define effect of moving function



Human\_being here inherited “Human” class under the default structure, so we need to rewrite some static functions



Define boxes in box.py



ATTITION: both of them inherited class from the default structure, so should add “from world.entity.entity\_import import \*” at begging of their file

**Generating the world**

Define functions to generate the defalut wolrd “ default\_generate\_a\_state()

“ in “world\_generator.py)

This function generate a world with size of 50\*50,

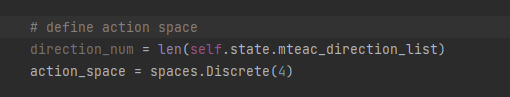
Default position of human is at (3,3)

There a two default boxes, (5,5) and (6,6)



Define action space

Because we have only for directions, so our action space is 4

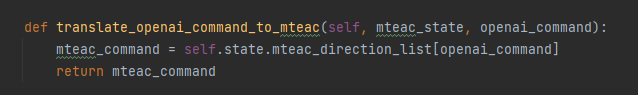


Define moving space

There are only block can be and can not be pass through, so moving space is a 2D array with a depth of 2.



Define translator



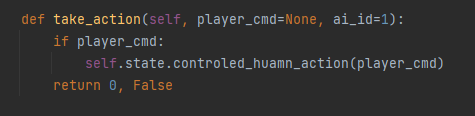
(We have not define “translate\_mteac\_state\_to\_openai” very well)

**Define logic between two rounds**

Moving function of human

This is the only functions that change states.

in “world.py”





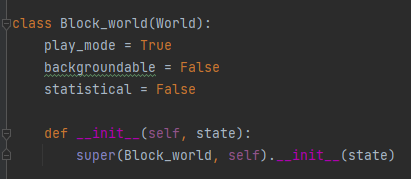
：

In “state.py”



**Simple visualization**

We manage to achieve “game mode” so we set “play\_mode” in “world.py” as 1



We import “pygame” library , use blocks to represent the world

White block can be pass through

Purple block can not be pass through

Green block is boxes

Red block is human

**Test result**

