

Game_test

Axect

August 18, 2016

1 3 Room Game

1.1 Test (동등한 관계의 방)

```
In [1]: # %%writefile Three_room.py

# -*- coding: utf-8 -*-

import random

class Room(object):

    door_list = ['Aroom', 'Broom', 'Croom'] # 일단은 모든 방 목록.

    def __init__(self):
        # 여기서는 모든방이지만 각각의 방에서 Override할 예정.
        # class의 variable은 클래스를 instance화 시키지 않아도 사용가능!
        self.door_list = Room.door_list

    def enter(self):
        print self.door_list # 어디로 갈 수 있는지 보여줌.

    def move(self):
        self.door = raw_input('>') # 일단, 텍스트로 움직일 방을 받음.

        if self.door in self.door_list:
            # 텍스트로 받은 방을 Map의 사전으로 방 객체를 호출 (이미 인스턴스화)
            self.next_door = Map.room_dict[self.door]

            print "%s으로 이동하였습니다." % self.door
            return self.next_door
        else:
            print "이동하지 않았습니다."
            return Map.room_dict[self.door]

class Engine(object):
```

```

def shuffle(self, x): # self는 클래스의 기본 인스턴스, x는 리스트
    self.x = list(x) # 즉, self.x란 Engine.x를 의미함.
    random.shuffle(self.x)
    return self.x

def init_point(self): # 시작위치를 랜덤으로 지정할것임.
    self.door_list = Room.door_list
    # Engine 내에 함수가 있기에 self.shuffle을 사용해야함. (self는 기본 인스턴스)
    self.shuffle_list = self.shuffle(self.door_list)
    return self.shuffle_list[0]

class Aroom(Room):

    door_list = ['Broom']
    def __init__(self):
        self.door_list = Aroom.door_list # Room의 init을 Override함.

class Broom(Room):

    door_list = ['Aroom', 'Croom']
    def __init__(self):
        self.door_list = Broom.door_list # Room의 init을 Override함.

class Croom(Room):

    door_list = ['Broom']
    def __init__(self):
        self.door_list = Croom.door_list # Room의 init을 Override함.

class Map(object):
    # Room class들을 인스턴스화시키고 사전에 저장.
    a = Aroom()
    b = Broom()
    c = Croom()

    # 텍스트로 받을 input을 객체로 변경시켜주기 위한 사전.
    room_dict = {
        'Aroom': a,
        'Broom': b,
        'Croom': c
    }

```

Overwriting Three_room.py

```

In [2]: from Three_room import *
        from time import sleep

```

```

def start():
    print "시작 위치를 지정합니다."
    E = Engine()
    init_point = E.init_point()
    print "시작 위치는"
    print "%s 입니다." % init_point
    print "%s 로 들어갑니다." % init_point
    # first_point 는 A,B,C 클래스 중의 하나를 객체화 시킨 객체.
    first_point = Map.room_dict[init_point]
    print "움직일 수 있는 방은 다음과 같습니다."
    first_point.enter() # 고로 enter를 상속받았음.

    for i in range(0,5):
        print "이동하시겠습니까? Y or N"
        OK = raw_input('>')

        if OK == 'Y':
            print "어디로 움직이시겠습니까?"
            # move도 상속되었기에 역시 가능. move의 반환값이 A~C의
            # 객체이므로 또 지정 가능.
            first_point = first_point.move()
            print "움직일 수 있는 방은 다음과 같습니다."
            first_point.enter() # 같은 이유로 enter도 상속됨.
        elif OK == 'N':
            print "편안히 쉬십시오."
            sleep(3)
            print "현재 위치는"
            first_point.enter()
        else:
            print "Error"
            break

start()

```

```

시작 위치를 지정합니다.
시작 위치는
Aroom 입니다.
Aroom 로 들어갑니다.
움직일 수 있는 방은 다음과 같습니다.
['Broom']
이동하시겠습니까? Y or N
>Y
어디로 움직이시겠습니까?
>Broom
Broom으로 이동하였습니다.
움직일 수 있는 방은 다음과 같습니다.
['Aroom', 'Croom']

```

이동하시겠습니까? Y or N
>N
편안히 쉬십시오.
현재 위치는
['Aroom', 'Croom']
이동하시겠습니까? Y or N
>Y
어디로 움직이시겠습니까?
>Croom
Croom으로 이동하였습니다.
움직일 수 있는 방은 다음과 같습니다.
['Broom']
이동하시겠습니까? Y or N
>N
편안히 쉬십시오.
현재 위치는
['Broom']
이동하시겠습니까? Y or N
>N
편안히 쉬십시오.
현재 위치는
['Broom']