

06/14/17 05:33:18 /home/zz/Documents/TIPE/src/graduator/IAGraduator.py

```
1  #!/usr/bin/python3.4
2  # -*-coding:Utf-8 -*
3
4  from math import ceil
5
6  from lib.inject_arguments import inject_arguments
7  from lib.inherit_docstring import inherit_docstring
8
9  from src.meta.ABCInheritableDocstringsMeta import ABCInheritableDocstringsMeta
10 from mario.bridge.config import Config
11 from mario.bridge.launch import launch
12 from src.EvolutiveGenerator.Graduator import Graduator
13 from src.entities.Result import Result
14
15
16 class IAGraduator(Graduator, metaclass=ABCInheritableDocstringsMeta):
17     """Graduate IA """
18
19     @inject_arguments
20     def __init__(self, event_dispatcher, show = False):
21         self.mario_x = 0
22         self.max_y = -500
23
24
25     def gradeIAWithConfig(self, ia, config):
26         # Init
27         self.mario_x = 0
28         self.max_y = -500
29
30         # Give the event_dispatcher to neurons
31         for neuron in ia.neurons:
32             neuron.event_dispatcher = self.event_dispatcher
33
34         self.event_dispatcher.listen('game.frame', self.onFrame)
35
36         # Launch game
37         persist = launch(config)
38
39         # Remove the event_dispatcher from neurons
40         for neuron in ia.neurons:
41             del neuron.event_dispatcher
42
43         # Make the result
44         result = Result(persist['camera start x'] + self.mario_x, self.max_y)
45
46         # Return the score
47         return result
48
49
50     @inherit_docstring
51     def grade(self, ia, generation_id):
52         time = 1 + ceil(generation_id / 2)
53         if time > 401:
54             time = 401
55
56         return self.gradeIAWithConfig(ia, Config(self.show, self.event_dispatcher, time))
57
58
59     def onFrame(self, frame):
60         self.mario_x = frame.mario.rect.x
61         self.max_y = max(self.max_y, - frame.mario.rect.y)
```