

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 523)

Equipo 5

Santiago Gabian Perez A01658280
Alan Alberto Flores Cuevas A01652453
Bruno Passarette Santos A01658904

Avance del código en python

```
9 class HRoadAgent(Agent):
10     def __init__(self, name, model):
11         super().__init__(name, model)
12         self.type = 1
13
14 class VRoadAgent(Agent):
15     def __init__(self, name, model):
16         super().__init__(name, model)
17         self.type = 3
18
19 class VTrafficAgent(Agent):
20     def __init__(self, name, model):
21         super().__init__(name, model)
22         state=0
23         self.type=4
24         self.state=state
25
26         self.running= True
27     def step(self):
28         self.state=1
29
30
31
32
33
34
35
36
37 class HTraffic(Agent):
38     def __init__(self, name, model):
39         super().__init__(name, model)
40         self.type=5
41         self.state=0
42     def step(self):
43         self.traffic(self.state)
44
45
46     def traffic(self, state):
47         state=1
48         self.state=state
49
50 class CarAgent(Agent):
51     new_position = (0,0)
52     """An agent with hopes, dreams, and a mysterious past."""
53     def __init__(self, unique_id, model):
54         super().__init__(unique_id, model)
55         self.type = 2
56         self.directionx=0
57         self.directiony=0
```

```
def step(self):
    self.move(new_position = (self.pos[0],self.pos[1]))

def move(self,new_position):
    n1=self.pos[0]+1,self.pos[1]
    n2=self.pos[0]-1,self.pos[1]
    n3=self.pos[0],self.pos[1]+1
    n4=self.pos[0],self.pos[1]-1

    if self.directionx==1 and self.model.grid.is_cell_empty
    (self.model.grid.torus_adj(n1)):
        new_position=(self.pos[0]+1,self.pos[1])

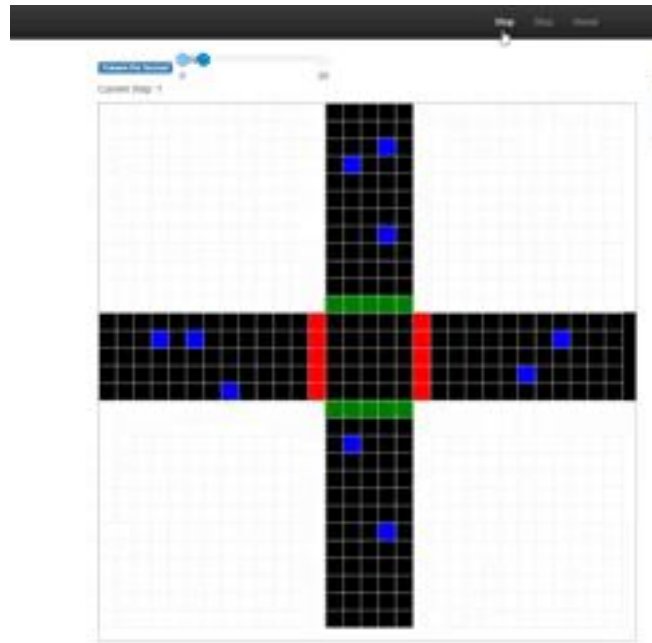
    if self.directionx==2 and self.model.grid.is_cell_empty
    (self.model.grid.torus_adj(n2)):
        new_position=(self.pos[0]-1,self.pos[1])

    if self.directiony==1 and self.model.grid.is_cell_empty
    (self.model.grid.torus_adj(n3)) :
        new_position=(self.pos[0],self.pos[1]+1)

    if self.directiony==2 and self.model.grid.is_cell_empty
    (self.model.grid.torus_adj(n4)) :
        new_position=(self.pos[0],self.pos[1]-1)
```

Visualización del funcionamiento

```
8 def agent_portrayal(agent):
9     portrayal = {
10         "Shape": "rect",
11         "Filled": "true",
12         "w": 1,
13         "h": 1}
14     if agent.type == 1:
15         portrayal["w"] = 31
16         portrayal["h"] = 5
17         portrayal["Color"] = "black"
18         portrayal["Layer"] = 0
19     elif agent.type == 2:
20         portrayal["Color"] = "blue"
21         portrayal["Layer"] = 1
22     elif agent.type == 3:
23         portrayal["w"] = 5
24         portrayal["h"] = 31
25         portrayal["Color"] = "black"
26         portrayal["Layer"] = 0
27     elif agent.type == 4:
28         if agent.state == 0:
29             portrayal["w"] = 5
30             portrayal["h"] = 1
31             portrayal["Color"] = "red"
32             portrayal["Layer"] = 1
33         elif agent.state == 1:
34             portrayal["w"] = 5
35             portrayal["h"] = 1
36             portrayal["Color"] = "green"
37             portrayal["Layer"] = 1
38     elif agent.type == 5:
39         if agent.state == 0:
40             portrayal["w"] = 1
41             portrayal["h"] = 5
42             portrayal["Color"] = "red"
43             portrayal["Layer"] = 1
44         elif agent.state == 1:
45             portrayal["w"] = 1
46             portrayal["h"] = 5
47             portrayal["Color"] = "green"
48             portrayal["Layer"] = 1
49
50     return portrayal
51
52 grid = CanvasGrid(agent_portrayal,
53                   30,
54                   30,
55                   800,
56                   800)
57
```



<https://bit.ly/3xp0vc8>

Plan de trabajo

Primera actualización

Introducción de los integrantes del equipo:

Responsables: Todos los integrantes cada uno realizó su introducción.

Entrega: 12 de Noviembre del 2021.

Duración: 15 minutos.

Descripción del proyecto:

Responsable: Santiago Gabian .

Entrega: 12 de Noviembre del 2021.

Duración: 20 Minutos.

Identificación de agentes:

Responsables: Todos los integrantes contribuyeron.

Entrega: 12 de Noviembre del 2021.

Duración: 30 Minutos.

Diagrama de clases:

Responsables: Bruno Passarete.

Entrega: 12 de Noviembre del 2021.

Duración: 10 Minutos.

Diagrama de protocolos:

Responsables: Alan Flores.

Entrega: 12 de Noviembre del 2021.

Duración: 40 Minutos.

Plan de trabajo

Segunda actualización

Diseño de coches e intersección en unity:

Responsables: Todos los integrantes contribuyeron.

Entrega: 18 de Noviembre del 2021.

Duración: 1 Hora.

Completo.

Avance de código de python:

Responsables: Todos los integrantes del equipo.

Entrega: 19 de Noviembre del 2021.

Duración del trabajo empleado: 3 Horas.

Status: Pendiente.

Expectativa de término: 3 Horas adicionales.

Plan de trabajo

Tercera actualización

Tercer avance de código de python (Solución 1):

Responsables: Bruno Passarete, Santiago Gabian

Entrega: 26 de Noviembre del 2021.

Duración del trabajo empleado: 10 Horas.

Status: Por acabar.

Expectativa de término: 1 Hora adicional.

Definición de materiales y assets a usar en Unity:

Responsables: Todos los integrantes del equipo.

Entrega: Sin fecha determinada.

Duración del trabajo empleado: 2 Horas.

Status: Pendiente.

Expectativa de término: 1 Hora adicional.

Tercer avance de código de python (Solución 2):

Responsables: Alan Flores

Entrega: 26 de Noviembre del 2021.

Duración del trabajo empleado: 2 Días.

Status: Por acabar.

Expectativa de término: 3 Horas adicionales.

Conexión base de datos IBM - Unity:

Responsables: Todos los integrantes del equipo trabajaron de manera individual.

Entrega: Sin fecha determinada.

Duración del trabajo empleado: 2 Horas.

Status: Completo.