Documentación de código

Alegandro Gallego, Sergio Pérez, Jhon Ramírez

Aqui va la fecha

1. Documentación

A continuación se presenta la descripción del comportamiento de cada una de las clases, así como el objetivo de cada uno de los métodos que las conforman, incluyendo sus parámetros de entrada y sus parametros de salida.

1.1. Agente

Representa un agente en el sistema. Está conformado por un componente de **Racionalidad**, y un componente de **Movilidad**, tiene la característica de dispersarse en la red, esto es, mover cada una de sus partes a otros nodos en la red, cada cierto intervalo de tiempo, determinado por una función de distribución de probabilidad.

```
1  import Pyro4
2
3  class Agente(object):
4
5      tipoMovilidad = ["constante","uniforme","exponencial"]
6
7      def __init__(self,nombre, movilidadId, racionalidadId, hostUri)
           :
8          self.hostUri = hostUri
9          self.nombre = nombre
10         self.movilidadId = movilidadId
11         self.racionalidadId = racionalidadId
12
13     def getMovilidadId(self):
14         return self.movilidadId
15
16     def getRacionalidadId(self):
17         return self.racionalidadId
18
19     def getNombre(self):
20         return self.nombre
21
22     def getType(self):
23         return 'head'
24
```

```
25      def getPyroId(self):
26          return str(self._pyroId)
27
28      def doIt(self):
29          ##place some call to legs and arms
30          racionalidadUri = Pyro4.Proxy(self.hostUri).resolve(self.
                racionalidadId)
31          movilidadUri =  Pyro4.Proxy(self.hostUri).resolve(self.
                movilidadId)
32          if (racionalidadUri == False or movilidadUri == False):
33              return 'Algo esta perdido'
34          racionalidad =x Pyro4.Proxy(racionalidadUri)
35          movilidad = Pyro4.Proxy(movilidadUri)
36          return [racionalidad.sayArms(), movilidad.sayLegs()]
```

Métodos

A continuación se presentan los métodos de la clase Agente, se proporciona una descripción del objetivo que se persigue en cada método, así como la descripción de sus parámetros de entrada, y sus parámetros de salida que retorna una vez termine su ejecución.

1.1.1. init(*nombre*, *movilidadId*, *racionalidadId*, *hostUri*)

Es el constructor de la clase, y a través de este método se permite instanciar objetos de esta clase.

```
1 def __init__(self,nombre, movilidadId, racionalidadId, hostUri):
2   self.hostUri = hostUri
3   self.nombre = nombre
4   self.movilidadId = movilidadId
5   self.racionalidadId = racionalidadId
```

Parámetros

- *nombre*: Nombre con el que se identifica el Agente en la red.
- *movilidadId*: Identificador del objeto que representa la parte correspondiente a la capacidad del agente de moverse en la red.
- *racionalidadId*: Identificador del objeto que representa la parte correspondiente a la capacidad del agente de tomar decisiones.
- *hostUri*: Identificador de recurso uniforme del *Host* donde está alojado el agente en un instante de tiempo específico.

1.1.2. getMovilidad()

Permite a los demás objetos en la red obtener el identificador del objeto de la movilidad del agente.

```
def getMovilidadId(self):
  return self.movilidadId
```

Retorno

Retorna el identificador del objeto que representa la parte correspondiente a la capacidad del agente de moverse en la red.

1.1.3. getRacionalidad()

Permite a los demás objetos en la red obtener el identificador del objeto de la racionalidad del agente.

```
def getRacionalidadId(self):
  return self.racionalidadId
```

Retorno

Retorna el identificador del objeto que representa la parte correspondiente a la capacidad del agente de tomar decisiones.

1.1.4. nombre

```
codigo
```

Parámetros

Retorno