



**Tecnológico
de Monterrey**

Modelación de sistemas multiagentes con gráficas computacionales

TC2008B.302

Revisión 2 - Modelación de Agentes

23 de noviembre 2022

Diego Araque | A01026037
Fernando Valdeón | A01745186
Marco Torres | A01025334

Agentes y su funcionalidad

Coches:

Estos recorren la ciudad en base a distintas reglas. Mientras el semáforo esté en verde y no tengan coches al frente estos podrán seguir moviéndose, pero si el semáforo está en rojo o hay una posibilidad de chocar estos se detendrán.

- **Atributos:**

- Posición (tupla) Indica la posición en la que se encuentra actualmente
- Ruta (arreglo tuplas) Indica la secuencia de celdas que debe seguir el auto.
- Mapa (grafo) Es el que nos permite calcular el camino más corto y recalcular si es necesario usando el algoritmo A*.

- **Métodos:**

- step: checa la posición actual, y llama una función respectiva de movimiento dependiendo de la dirección a la que debe moverse.
- move(direction: string): mueve el coche.
- getNearestPath: utiliza el algoritmo a* para calcular el camino más cercano a un destino.
- getDistanceToDestination(destiny: tupla): calcula la distancia euclidiana entre la posición actual del vehículo y un destino.
- isPathBlocked(): revisa 5 posiciones enfrente del vehículo y regresa True si se encuentran 5 vehículos frente a él, esto con el fin de poder tomar la decisión de cambiar de ruta .
- isTrafficLightRed(): revisa la casilla de enfrente y si existe un semáforo que además esté en rojo regresa true, de lo contrario false.
- checkIfCarInFront(): revisa si tiene un coche al frente, si si este no se mueve

Semáforos: Cambian entre rojo y verde, esto dará el comportamiento que queremos a los coches.

- **Atributos:**

- Estado (boolean) True para verde False para Rojo
- Posición (tupla)

Calles:

Estos no tendrán un comportamiento fijo, en lo único en lo que influyen es que los coches recorren la ciudad sobre ellos y se ubicarán por medio de coordenadas.

- **Atributos:**

- Dirección (string). Indica el sentido de la calle.
- Disponibilidad (boolean). Indica si está ocupada o no
- Posición (tupla)

Edificios:

Estos servirán como obstáculos, por lo tanto no tendrán interacción con ningún otro agente dentro del modelo.

- **Atributos:**

- Posición (tupla)

Puntos de llegada:

Estos son unos de los más importantes, en nuestra ciudad se encuentran algunos puntos de llegada que serán edificios a los cuales tienen que llegar los coches. Los coches deberán llegar tomando una ruta rápida.

- **Atributos**

- Posición (tupla)

Algoritmo e interacción entre agentes

Inicialmente tendremos un grid con los agentes de obstáculos, caminos y destinos, nuestros carros aparecerán en el grid desde las esquinas. Cuando un coche aparezca el primer paso que debe de realizar es obtener la ruta más corta hasta su destino. Para esto, implementaremos el algoritmo A*, el cuál deberá tener en consideración los sentidos de los caminos, y deberá de regresar la secuencia de celdas que debe de seguir el coche.

El coche empezará a seguir el camino indicado, frenando cuando haya otro auto en frente o se encuentre con un semáforo en rojo. El coche también podrá determinar si su ruta actual asignada está bloqueada por tráfico, en ese caso deberá recalcular la siguiente ruta más corta, ahora tomando en cuenta los coches que causan el embotellamiento como obstáculos. Al recalcular la ruta si no encuentra algo más eficiente, nos mantendremos en la misma, de otra forma se busca que el coche cambie de carril para acelerar el proceso. Cuando llegan al punto de destino, desaparece el coche y cumple con su objetivo.

Cada 20 pasos, aparecerán nuevos coches en cada esquina de la ciudad. Como se explicó anteriormente, apenas aparecen los coches estos tendrán su ruta ya calculada sin tomar en cuenta ningún tipo de embotellamiento. Y seguirá el mismo proceso para llegar al punto de destino.