Tecnológico
de Monterrey

Jesús Palomino Hurtado (A01638492)

Instituto Tecnológico y de Estudios Superiores de Monterrey

"Actividad Integradora"

Modelación de sistemas multiagentes con gráficas computacionales

22 de noviembre de 2021

Para poder representar el entorno del cruce de un paso peatonal en una avenida de dos carriles se creó una topología espacial bidimensional de N*M casillas utilizando la propiedad de Space dentro de la librería de AgentPy de Python. Cada casilla dentro de la topología creada representa una posición dentro del entorno, sobre las cuales se posicionarán y desplazaran los diferentes agentes y objetos dentro del entorno, en caso de que un agente se intente desplazar a una posición inexistente dentro del espacio definido, este es posicionado en la casilla con valor mínimo o máximo del mismo eje del cual se intenta poner un valor no válido dependiendo respectivamente de si se intenta poner en un valor mayor o menor a los definidos dentro del espacio.

Dentro del entorno, un conductor se puede enfrentar a diferentes situaciones las cuales dependen de diferentes variables de los agentes de su alrededor, estas situaciones son:

1. El conductor choca con otro carro
2. Hay un carro adelante y está a menos de 70 metros
3. Hay un carro adelante y está a más de 70 metros y menos de 150 metros
4. Semaforo del carril esta adelante, está en amarillo y la distancia entre los dos es menor a 60 metros
5. Semaforo del carril esta adelante, está en amarillo y la distancia entre los dos es mayor a 60 metros y menor a 120 metros
6. No hay un carro adelante, tiene velocidad menor a 5 y el semáforo del carril esta adelante, en rojo o amarillo y esta a más de 60 metros
7. No hay un carro adelante, tiene velocidad menor a 5 y el semáforo del carril está adelante, en rojo o amarillo y esta a menos de 60 metros
8. Semáforo del carril en rojo y a menos de 200 metros
9. Cualquier otra situación

Para cada una de estas situaciones se tiene una respuesta diferente por parte de un conductor, estas son las acciones para cada situación respectiva:

1. El carro ya no se puede mover de su posición
2. Frenado rápido hasta llegar a una velocidad de 0 o encontrar otra situación
3. Frenado lento hasta llegar a una velocidad de 0 o encontrar otra situación
4. Acelerar poco hasta llegar a la velocidad máxima establecida o encontrar otra situación
5. Frenar poco hasta llegar a una velocidad de 0 o encontrar otra situación
6. Acelerar poco hasta llegar enfrente del semáforo o encontrar otra situación
7. Frenado rápido hasta llegar a una velocidad de 0 o encontrar otra situación
8. Frenado moderada hasta llegar a una velocidad de 0 o encontrar otra situación
9. Acelerar poco hasta llegar a la velocidad máxima establecida o encontrar otra situación

Para simular el entorno y su interacción, se utilizó un enfoque de agentes dentro de Python con la librería de PyAgent, dentro de la cual se definió el modelo como la avenida y los agentes carro y semáforo, los cuales se pueden observar dentro de la *figura 1*. Se recopiló la información del estado del equivalente de cada paso de 0.25 segundos dentro de la simulación en un archivo JSON, esta información fue recopilada sobre los diferentes agentes dentro de la simulación, tales como sus posiciones, orientaciones, estados e identificadores.
Además de esto, dentro de la simulación también se reportaron datos como la cantidad de automóviles que chocaron y se realizaron pruebas de diferentes situaciones, tales como si el tiempo en el que se muestra la luz amarilla de los semáforos se reduce a 0.

En el caso de la simulación base, la cual es generada por el archivo de python sin alguna modificación, la cantidad de autos chocados fue de 0, en cuanto a la situación el la que el tiempo de duración de la luz amarilla se redujo a 0, la cantidad de autos chocados tambien fue de 0, sin embargo los carros presentaron una frenado más brusco para parar en frente del semáforo.

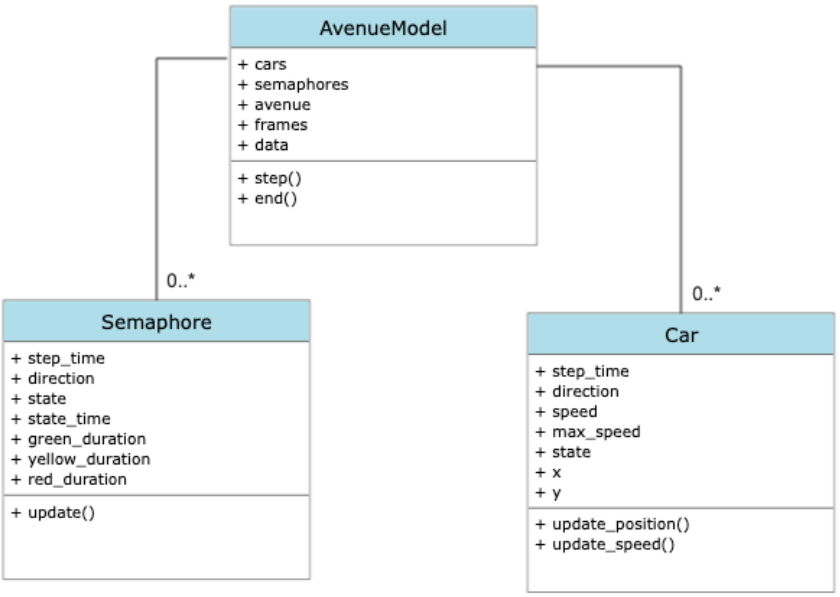

Figura 1. Diagrama de clases de modelo de simulación