



Reto: Movilidad Urbana

Germán Andrés Jaramillo Ramírez - A00571636

Esteban Sánchez García - A01251440

Ernesto Adrián Álvarez Salazar - A00227490

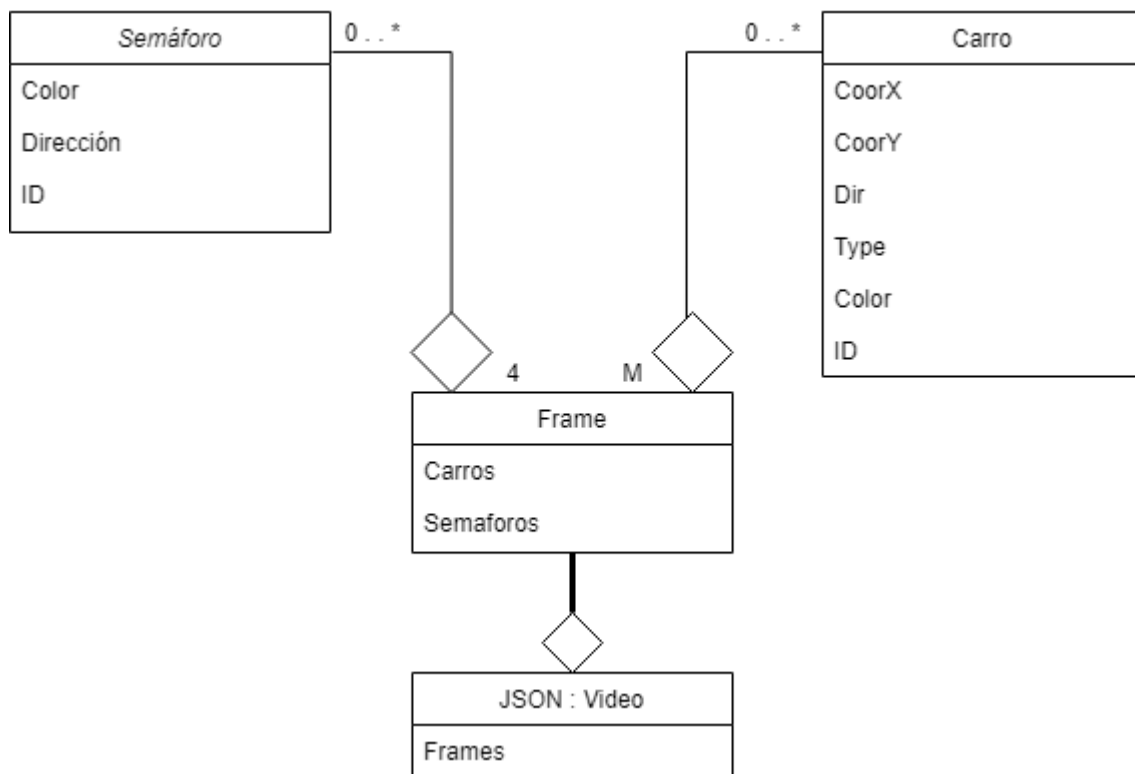
Nickolás Rodríguez Ochoa - A01251446

Javier Lizarraga Beyles - A01253105

30 de noviembre de 2021



Diagrama de clases:



Protocolos de interacción finales:

Primeramente se toma la decisión de los agentes representando semáforos, los cuales tienen una lógica distinta entre ellos:

- Aleatorio: Este semáforo escoge aleatoriamente cada votación el orden de prioridad.
- Aleatorio singular: Escoge aleatoriamente la primera vez, y se queda con esa selección.
- Egoísta: Se pone la mayor prioridad a si mismo, y deja a los demás con la segunda menor prioridad.
- Por necesidad: El último escoge a quien tenga mayor cantidad de carros, en caso de empate quien tenga prioridad de índice.

Los semáforos votan y cada quien da su lista de prioridad, se suman las prioridades y se saca una nueva lista de prioridad comunitaria, y después que se cumple la lista 1 vez se vuelve a votar y así sucesivamente. La votación decide quién estará encendido en verde, y cuando llega a rojo de nuevo sigue el siguiente semáforo. Solo se prende un semáforo a la vez para poder llevar mayor escalabilidad y poder implementar en un futuro el semáforo de vueltas.

Los siguientes agentes, los cuales dependen del tipo de agente semáforo, son los autos. Sus protocolos son los siguientes, avanzan a su velocidad asignada, con una determinada velocidad maxima y minima general, hasta encontrarse con un carro frente a ellos a 60 metros de distancia por lo que reducirá su velocidad hasta 2 si es necesario, y si esta a menos de 40 metros entonces si opta por el frenado total, si no hay carro frente a él se detendrá dependiendo de el estado del semáforo, si es

“verde” cruza sin reducir la velocidad, si está en “rojo” se pone en un alto total hasta llegar a una distancia de entre 40 y 42 metros, pero si esta a menos de 40 metros optó mejor por pasarselo, cuando el estado es “amarillo” pueden suceder tres cosas: si esta a menos de 30 metros cruza sin reducir velocidad, si esta entre 30 y 60 metros trata de frenar hasta llegar a una velocidad de 5, y si esta a mas de 60 empieza a acelerar para tratar de tener mucha velocidad y así que el frenado no sea suficiente para detenerse este el semáforo en cualquier estado.

Durante el proyecto fuimos iterando y cambiando los parámetros de la velocidad máxima, aceleración y el frenado de los carros, hasta que llegamos a un punto más adecuado. Al principio optamos por mayor aceleración, velocidades muy altas y frenados muy agresivos, pero al correr la simulación se veía rara en general, por lo que mejor reducimos la velocidad máxima, reducimos el freno pero aumentamos ligeramente la distancia mínima, etc.

Proceso de instalación y configuración:

Como primer paso se clonara el repositorio de github:

https://github.com/JavierLizarragaB/Actividad_Integradora_Multiagentes

En el cual obtendrás varios archivos, los cuales no se necesitan modificar, a continuación se instalarán las dependencias de la simulación de multiagentes: numpy, agentpy.

Se deben correr las siguientes líneas en terminal (en esta parte se considera que ya se cuenta con python y pip):

```
pip install numpy
pip install agentpy
```

Si se desea configurar los parámetros, se debe modificar la sección de “parámetros”, que se encuentran aproximadamente en la línea 442 del archivo agentes.py, en donde se pueden ver las distintas variables que afectan la simulación:

- `step_size`: el tiempo que dura cada frame de la simulación
- `size`: no debe cambiarse pues podría causar imprevistos en la vista gráfica de la simulación, pero si se cambiase podría hacer el espacio de la simulación más grande o pequeño. Cabe mencionar que este es el tamaño de un lado y el tamaño final sería este valor al cuadrado.
- `green`: tiempo que dura el semáforo en verde
- `yellow`: tiempo que dura el semáforo en amarillo
- `cars`: numero de carros que entran en la simulación
- `steps`: número de frames que tendrá la simulación

Tras esto se ejecuta el archivo de python de la siguiente manera:

```
python3 agentes.py
```

Esto generará un archivo json con la posición y el estado de todos los agentes en cada frame, la cual se utilizara en el index.html que abrirán en su navegador mostrando así la simulación gráficamente.

Liga de vídeo demostrativo de la instalación:

https://drive.google.com/file/d/1MKwUL48VIHYDjSiqggmj_Jyb3gxNCjkO/view?usp=sharing