

¿Por qué seleccionaron el modelo multiagentes utilizado?

Para la simulación utilizamos el framework de Mesa, ya que es el que se pudo encontrar mayor documentación y ejemplos de como utilizarlo. Nos permite desarrollar varios agentes y ver la interacción que tienen entre ellos. Se puede modificar el modelo como nosotros querramos para simular la situación que deseemos y no solo eso, este paquete viene incluido con un recolector de datos el cual nos permite analizar con mayor profundidad el comportamiento de lo que modelemos, esto es bastante util a la hora de recoger datos y querer optimizar nuestro modelo.

El reto que debíamos modelar podía modelarse bastante bien con todas las funciones que Mesa incluye, un simple grid que representa la ciudad y los agentes son nuestros vehículos, semáforos, edificios, etc.

¿Cuáles fueron las variables que se tomaron al momento de tomar la decisión?

Creo que lo más importante fue que hubiera una documentación adecuada y detallada del framework para poder referirnos a ella cuando lo necesitáramos. También se trató de plasmar el mismo reto con distintos modelos y seleccionamos el que mejor nos podía ayudar de resolverlo eficientemente.

¿Cuál es la interacción de esas variables con respecto al resultado de la simulación?

Al final se pudo resolver el problema con una buena implementación en Mesa. Se puede observar el comportamiento deseado de los vehículos en el servidor que provee este framework.

¿Por qué seleccionaron el diseño gráfico presentado?

Se nos entregaron los assets de la ciudad y los semaforos a utilizar, nosotros escogimos el de los vehículos. Escogimos coches que no tuvieran una cantidad demasiada alta de polígonos ya que ese simple modelo 3D se volvería muy pesado, y a la hora de querer simular nuestra implementación con varios coches, se volvería imposible de observar lo que esta sucediendo. Particularmente tambien tuvimos problemas con las rotaciones de otros vehículos, hasta que llegamos a uno donde podía su posición era correcta.

¿Cuáles son las ventajas que encuentras en la solución final presentada?

Primero que es un modelo en el cual se pueden observar los comportamientos de los agentes, se puede ver como interactúan entre ellos y con el ambiente y respecto a eso ir mejorandolo y corrigiendo errores y bugs con mayor facilidad. También que se pueden recoger datos mientras corre la simulación y presentarlos en formas de gráficos. Esto es muy útil si se requiere hacer un

análisis de datos, y el hecho de que este implementando en Python permite usar al mismo tiempo otras librerías para llegar a conclusiones sobre lo que esta pasando, como Numpy, TensorFlow, SciPy, Pandas, etc.

¿Cuáles son las desventajas que existen en la solución presentada?

El comportamiento de los agentes podría ser mejor si el framework tuviera funciones implemtnadas para tener comunicación directa con otros agentes. Esto lograría algo más natural y nos daría la posibilidad de simular distintas situaciones, como por ejemplo, que pasa si dos coches chocan, o uno se pasa el alto, o uno nunca llega a su destino.

¿Qué modificaciones podrías hacer para reducir o eliminar las desventajas mencionadas?

Creo que nosotros tendríamos que pensar, plantear diferentes situaciones que suceden en la vida real para que tambien pasen en la simulación y los agentes reaccionen a estas. También es necesario más tiempo para implementar todo esto, ya que salen muchos errores y bugs que consumen muchas horas.