

Análisis de la Solución

El reto planteado para este bloque fue simular el tráfico de una ciudad donde se encuentran varios vehículos con objetivo de llegar a su destino y varias calles incluyendo intersecciones con semáforos. Para la solución del reto se utilizó un módulo de Python llamado 'Mesa' que permite la construcción y visualización de modelos multiagentes. Con este módulo pudimos desarrollar un sistema multiagente representativo del tráfico en una ciudad. Todo esto por medio de la definición del modelo y los agentes que actúan dentro del modelo. La parte de visualización se realizó en Unity haciendo una conexión con un servidor de 'flask' que es otro módulo de Python que permite la creación de aplicaciones web. Este servidor fue implementado para que pueda ser ejecutado en la nube de IBM o de manera local.

Por la manera que funciona 'Mesa', se definieron agentes que representan los destinos, las celdas de calle, las celdas que son obstáculos (edificios) y los semáforos, para darles funcionalidad a estos objetos presentes en el modelo, aunque técnicamente no son agentes. El agente que verdaderamente tiene un comportamiento informado es el carro, que de la misma manera fue definido con base a un agente del módulo de mesa, pero obviamente con más funcionalidad que los objetos mencionados previamente.

Antes de programar todo esto, tuvimos una sesión de equipo donde analizamos todas las variables a considerar para construir de manera completa y eficiente lo que se plantea en el reto. Durante esta sesión construimos el diagrama de jerarquía de decisiones para el agente, donde consideramos los momentos donde el carro tiene que detenerse y cuando se puede mover para que actúe de manera racional dependiendo de su entorno y su estado actual. Las variables que influyen en esta decisión fueron identificadas como: el destino del carro, la dirección de la calle donde se encuentra actualmente el carro incluyendo intersecciones, objetos presentes enfrente del carro como otro carro o un semáforo y por último el estado actual del carro ya sea en movimiento, detenido o si ya llegó a su destino. Durante esta sesión también definimos los diagramas de clases y con estas bases completas pudimos avanzar a la programación del sistema multiagentes.

Lo primero que implementamos considerando las variables descritas previamente fue el algoritmo A* para encontrar el camino más corto hacia el destino de cada carro, lo cual resultó un poco más retador de lo pensado ya que tuvimos que considerar cada situación presente en la base de la ciudad y las direcciones de las calles presentes. Pero así, además de encontrar el camino, ya toma en consideración la dirección de la calle. Después de que el carro ya tenga el camino hacia su destino, es cuestión de que siga ese camino pero siempre siguiendo las reglas del tráfico y evitando colisiones con otros carros.

Para que el carro tenga buen cumplimiento de las reglas de tráfico, que en este caso es parar en semáforos rojos y no chocar con otro carro, antes de dar el siguiente paso el carro tiene que analizar la celda hacia dónde se quiere mover. Si la celda es un semáforo que está en rojo o tiene un carro que no está en movimiento (viendo su siguiente paso), el carro le da prioridad a detenerse en vez de seguir su camino para así evitar colisiones. Finalmente, cuando el carro llega a su destino, es marcado como estacionado para que no siga analizando su entorno en este estado. La simulación acaba cuando todos los carros llegan a su destino.

Al tener todo lo previamente mencionado implementado correctamente pudimos ver que la interacción entre estas variables definidas, cuando implementadas con la prioridad adecuada, resultan en una simulación de una ciudad donde los carros pueden llegar a su destino en el camino más corto, siempre evitando colisiones y respetando las luces de los semáforos.

¿Por qué seleccionaron el diseño gráfico presentado?

Para la parte de la visualización del modelo, decidimos usar Unity ya que fue la herramienta utilizada en proyectos de semestres pasados por lo cual facilitó aprender sobre modelación e iluminación en una plataforma donde ya teníamos experiencia. Además de esto también teníamos experiencia en la creación de una conexión para la comunicación entre unity y nuestro API, lo cual resultó bastante útil por el tiempo limitado disponible en este bloque de 5 semanas.

Con respecto a los modelos gráficos, utilizamos un modelo de carro y otro de semáforo hecho por uno de los integrantes del equipo. Estos modelos son de baja resolución, lo que significa que no tienen muchos vértices lo cual resulta en cálculos de rotación y transformación más eficientes,

como vimos en clase. Además, ocupan menos tiempo en desarrollar lo cual permitió dedicarle más tiempo a la creación del sistema multiagente en ‘Mesa’.

Una de las ventajas de nuestro modelo es que utiliza el scheduler “Simultaneous activation” que permite simular un comportamiento simultáneo para los agentes ya que primero calcula los pasos de todos los agentes y después los aplica. Esto permite que el carro pueda saber el siguiente paso de un carro en su entorno y resulta en un movimiento más eficiente entre los carros.

Considero que nuestra implementación del algoritmo A* para encontrar el camino entre el carro y su destino como otra de las ventajas de nuestro modelo ya que es una manera eficiente de encontrar el camino entre el carro y su destino, solo lo tiene que calcular una vez al principio de la simulación y funciona en armonía con el resto de las prioridades del agente.

Una de las desventajas en la solución presentada es que en la visualización gráfica en Unity usando la conexión con el API en la nube los movimientos se ven trabados de vez en cuando porque llegan los ‘responses’ del API un poco lentos ya que estamos utilizando una instancia básica gratuita. Para resolver esto, se puede correr de manera local y la animación de los movimientos se ve más suave, pero una mejor solución sería crear una instancia en la nube con mejor conexión para que el response time sea menor.

Reflexión

Durante el transcurso del bloque pude adquirir aprendizaje sobre la modelación de sistemas multiagentes y un poco sobre gráficas computacionales y cómo funcionan. En las expectativas había dicho que quisiera aprender sobre inteligencia artificial, lo cual siento que no fuimos tan profundo en esta clase, comprensiblemente por el tiempo limitado que tuvimos. A pesar de esto considero que fue un bloque con mucho aprendizaje que servirá como base para futuros bloques y experiencias. En las expectativas también mencione que Inteligencia Artificial es una de mis areas de interés y este bloque me deja con ganas de aprender aún más sobre el tema.