



**Tecnológico
de Monterrey**

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 4)

Reto: Movilidad Urbana (Documento individual)

David Josué Marcial Quero

Ingeniería en Tecnologías Computacionales (ITC)

Campus Monterrey

Análisis de la solución desarrollada

- Selección del modelo multiagentes utilizado

La razón detrás de seleccionar un modelo de multiagentes para representar la intersección vehicular fue debido a que, como visto en clase, los agentes racionales permiten desarrollar sistemas que son razonablemente inteligentes o que actúan de manera racional. Puesto que queríamos tener una intersección eficiente y que pueda funcionar de forma dinámica, tener una modelación basada en agentes parecía la mejor opción, puesto que un agente realmente puede ser cualquier cosa que es capaz de percibir su entorno o ambiente a través de sensores, y actuar en dicho entorno.

Hablando en términos más específicos, dado que se quiere optimizar el cruce de vehículos en una intersección, es importante la implementación de agentes racionales, pues este es aquel que emprende una acción que maximiza su medida de rendimiento (en este caso cruzar una intersección sin colisiones, de forma ordenada y eficiente), basándose en las evidencias aportadas por la secuencia de percepciones (la detección de vehículos o banquetas próximas) y en el conocimiento que el agente mantiene almacenado (como la dirección del vehículo, sus coordenadas, etc).

Evidentemente, cuando quisimos modelar los vehículos, comenzamos a hablar de agentes reactivos basados en modelos, pues estos son los agentes que realizan un seguimiento del entorno a través de un modelo interno con el cual el vehículo conoce la evolución de sí mismo de acuerdo con las acciones realizadas. Como ya se mencionó en el párrafo anterior, es de vital importancia almacenar información de las distintas partes que conforman el escenario (en este caso una intersección automovilística), pues esto le permite reaccionar dependiendo de la "historia" percibida y que de ese modo refleje alguno de los aspectos de su estado actual (dirección, detenido o no, etc.).

Tenemos, en esta misma modelación, dos agentes basados en objetivos, pues estos solo contienen la información del objetivo que describe las situaciones que son deseables. Además, contienen la descripción de su estado actual, la cual es utilizada por el agente del vehículo para interactuar con el escenario y tomar decisiones informadas.

Al deliberar cómo sería el modelo, en cuanto a su representación en una estructura de datos para ser pasado a Unity, llegamos a la siguiente conclusión: utilizar una cuadrícula con espacio toroidal, el cual nos permita no destruir agentes una vez que llegan al final de una calle y solo que estos aparezcan del lado contrario de la cuadrícula. Por lo tanto, no se logra un apego tan realista como nos

hubiera gustado, sin embargo, esta decisión se tomó para facilitar la programación de esta modelación.

- Variables tomadas en la elección del modelo

Las variables que se tomaron en cuenta para decidir y diseñar el modelo fueron las siguientes:

- Vehículo: Este es el elemento de acción más importante dentro de la modelación, pues este fue sobre el cual se requirió un mayor cuidado en el diseño, así como la toma de decisiones más fundamental.
- Banqueta: Como se estipuló en el diseño del modelo, se trata de una variable que sirve para ayudar al vehículo a delimitar su área de movimiento.
- Carretera: De forma similar a la variable anterior, esta variable sirve para ayudar al vehículo a delimitar su área de movimiento, con la notable diferencia de que habrá diferentes tipos de carretera (normal, inmóvil y debajo de semáforo).
- Semáforo: Al igual que con la variable de los vehículos, el semáforo es uno de los elementos más importantes, pues este le permite saber al vehículo si es posible pasar por la intersección o no.

- Interacción de las variables en la simulación

La interacción que tuvieron las variables de la simulación primeramente estaría establecida por una comunicación de agentes con un lenguaje similar al de KQML, pues se utilizó un lenguaje de consulta y manipulación de conocimientos, lo cual es justamente lo que se lleva a cabo en la interacción de los agentes vehículos y semáforos. Por dar un ejemplo de esto, el vehículo consultará en cada turno si la carretera sobre la que se encuentra es una parada (donde hay un semáforo) y también se comunicará con el semáforo para saber si este está en rojo, amarillo o verde.

- Razón del diseño gráfico presentado

La razón por la que diseñamos la representación gráfica de la simulación en unity con modelos 3D de cada una de las variables que interactúan en dicha simulación, fue porque esto nos permite ver con mayor claridad y comprensión el resultado del modelado realizado con *MESA*.

Hablando de lo particular, creo que uno de los resultados más satisfactorios de la representación 3D del modelo fue que podemos visualizar cómo es que los vehículos cambian de dirección por medio de una rotación del modelo 3D. De igual

forma, se puede observar la iluminación de los diferentes colores que tienen los semáforos y cómo es que esto se relaciona con el paso de los vehículos.

- Ventajas de la solución presentada

Las evidentes ventajas de la solución presentada son las siguientes:

- Ninguno de los vehículos colisiona con otro.
- Los vehículos, al llegar a la intersección, pueden ir derecho, hacia a izquierda y derecha (desde el punto de referencia de cada vehículo).
- Tiene un comportamiento realista, es decir, cada carretera y, por lo tanto, cada semáforo tiene un tiempo determinado para dejar pasar a los vehículos de una cierta carretera.
- El semáforo, al no tener ningún vehículo próximo se pondrá en color amarillo. Este comportamiento permite que, aunque aún haya tiempo restante para estar en verde, pero no hay ningún automóvil que realmente vaya a pasar, se le permita el paso a otros vehículos de otras carreteras que sí quieren cruzar la intersección.

- Desventajas de la solución presentada

- En teoría, se podría tener una intersección en la cual se permita tener hasta cuatro semáforos en verde, pues podría haber un caso en el que los vehículos más cercanos al semáforo no tomen una dirección de colisión con los otros tres vehículos. Por lo tanto, es una desventaja no tener esta funcionalidad, pues de ser así, se tendría una modelación mucho más eficiente.
- Se tiene un tiempo establecido para mantener un semáforo en verde, por lo cual, independientemente si llegan más vehículos a una línea, pero el tiempo se ha terminado, no podrán pasar. Esto refleja un modelo apegado a la realidad, sin embargo, se podría tener que mientras sea el turno de este semáforo en particular y sigan llegando carros, se continúe estando en verde.

- Modificación para reducir las desventajas

Como se explicó brevemente en las desventajas, creo que una de las mejores formas de mitigar las desventajas y, en general, mejorar el modelo para hacerlo lo más eficiente posible para todos los vehículos, sería que la intersección y los semáforos funcionen de forma similar a una intersección 1X1. Es decir, el primer vehículo que desea pasar por la intersección lo hará y luego será seguido por el vehículo que se encuentre en otra parte de la intersección e igual desee pasar. No obstante, solo esto no generaría una mejora, sino que incluso sería menos

eficiente que la modelación actual. Por lo tanto, se tiene que modificar un poco más, de tal forma que se permita el pase simultáneo de vehículos. Además, solo se debe permitir que los vehículos que posean direcciones que no provoquen colisiones con los otros vehículos, por lo cual se debería crear un algoritmo que le permita a los semáforos identificar y calcular qué vehículos de qué carreteras permitir pasar, de forma tal que la mayor cantidad de vehículos pase por la intersección.

Reflexión sobre el proceso de aprendizaje

Al reflexionar un poco sobre los aprendizajes obtenidos durante el transcurso de este bloque, realmente creo que me llevo los conocimientos fundamentales para poder incursionar en las aplicaciones de inteligencia artificial de agentes racionales. Verdaderamente me he llevado una buena impresión de lo útiles que pueden ser para modelar problemas que, resueltos de otra forma, se volverían ineficientes o simplemente no lo suficientemente detallados. Además, me queda claro que estos logran ser una gran mejora a implementación o modelaciones tradicionales por el hecho de que, por cada mejor acción posible, el agente obtiene una “recompensa” positiva y por cada acción incorrecta, este agente obtiene una recompensa negativa.

De igual forma, al realizar la visualización de la modelación, creo que aprendí lo valioso que puede ser el tener una presentación idónea para una dada modelación o implementación teórica. El aplicar los conocimientos de modelación y transformaciones 3D también me hicieron crecer en mi entendimiento de la complejidad de las gráficas computacionales y de las distintas formas en las que en la actualidad se pueden realizar y mostrar, ya sea por un motor gráfico, tanto de forma programática como de forma interpretativa, así como directamente programar hacia el GPU por medio de CUDA.

En conclusión, creo que logré obtener los aprendizajes que me propuse al inicio del bloque, dado que realmente deseaba poder entender la renderización de objetos 3D en tiempo real, así como entender y practicar el modelado de sistemas de la vida real utilizando agentes. Evidentemente, aún hay mucho más que puedo aprender y mejorar en mi entendimiento de estos temas, sin embargo, creo que lo que he logrado entender me ayudará en demasía en el futuro.