



Tecnológico de Monterrey

Campus Santa Fe

Reto: Movilidad Urbana

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 302)

Alumnos:

Valentina Castilla Melgoza A01028209

Darío Cuauhtémoc Peña Mariano A01785420

Profesor:

Gilberto Echeverría Furió

Octavio Navarro Hinojosa

Viernes 5 de diciembre del 2025

El problema que buscamos resolver consiste en diseñar una simulación donde múltiples agentes (en nuestro caso, huevos) sean capaces de llegar a sus destinos en el menor tiempo posible mientras respetan todas las reglas del entorno. Para esto, utilizamos el mapa dado, donde se especifica la estructura del ambiente incluyendo calles, sentidos de estas, semáforos y obstáculos. A partir de estos elementos, construimos el ambiente en el que los agentes se desplazan.

Los agentes solo pueden transitar en el sentido permitido por cada calle, por lo que la simulación debe garantizar que se respeten los flujos de tráfico y las restricciones. Además, cada 10 *steps* se agregan nuevos agentes en cada una de las esquinas, lo que incrementa la posibilidad de congestiones. El objetivo principal de cada agente es alcanzar un destino específico. Una vez que lo logra, el agente desaparece del ambiente para mantener el flujo de la simulación. Para desplazarse, los agentes deben evitar colisiones con otros agentes, esquivar obstáculos, respetar los semáforos y buscar no generar congestiones innecesarias.

Para esta simulación se trabajó con un mapa compuesto por símbolos. Cada símbolo representa un objeto físico del entorno como calles con distintos sentidos, obstáculos y lugares destino. Estos símbolos también están vinculados con agentes que cumplen diferentes funciones. El agente más complejo es el agente huevo, cuyo objetivo es desplazarse desde su punto de aparición hasta su destino en el menor tiempo posible. Para lograrlo se diseñó una estrategia basada en un grafo que se describe a continuación.

Con el objetivo de generar rutas óptimas se definieron nodos en cada intersección del mapa. Cada nodo incluye conexiones hacia otros nodos a los que es posible llegar según las direcciones de las calles cercanas y las reglas de tránsito aplicables. Esta estructura permitió construir un grafo dirigido que conecta las intersecciones del mapa con los puntos destino, considerados como nodos finales.

Una vez creado el grafo se empleó un algoritmo de búsqueda por anchura para determinar las direcciones que debe seguir el agente huevo desde su origen hasta su destino. Con esto cada agente dispone de una ruta definida para alcanzar su objetivo. Además se dotó al agente de capacidades para respetar las leyes de tránsito y para rebasar en diagonal a obstáculos y a otros agentes que puedan aparecer en su recorrido. El diseño de este agente es el siguiente

Diseño de los agentes

Agente Huevo

Objetivo: Seguir una ruta de punto de aparición a destino respetando las direcciones de las calles y leyes de tránsito, evitando objetos y otros agentes en el camino para no chocar y producir tráfico.

Capacidad efectora:

- Seguimiento de rutas en base a intersecciones en el mapa.
- Rebase y rodeo de otros agentes y obstáculos para evitar chocar.
- Respetar las reglas de tránsito .
-
- Detenerse por semáforo en rojo o por tráfico en el entorno

Percepción: Los agentes sólo pueden conocer su estado y lo que está pasando alrededor de ellos, además de percibir las direcciones de las calles, si hay otros huevos en otras celdas, los estados de los semáforos y detectar cuando está en una intersección o en un destino.

Proactividad:

- Los agentes tienen la iniciativa de rebasar cuando detecta un obstáculo enfrente y detecta que la celda para rebasar está vacía.
- Los agentes definen rutas alternativas cuando la ruta para llegar al destino está bloqueada.
- Los agentes planean su comportamiento para llegar a su destino

Métricas de desempeño: Para esta simulación se tomaron 3 métricas de desempeños para saber el rendimiento, las cuales son huevos en pantalla, que son los huevos que existen en la simulación, huevos totales que han aparecido y el número de huevos que han llegado a su destino.

Habilidad social: Para este caso en específico, los agentes no cuentan con habilidad social, ya que otros agentes no podían podían conocer los estados de otros, lo único que conocen de es la información inmediata de los agentes vecinos.

Arquitectura de subsunción de los agentes:

La arquitectura del agente se dividió en capas, lo que permitió desarrollar y obtener cierta información de manera modular, las capas son las siguientes:

Capa 1: Evitar choques

Es el comportamiento básico y tiene la mayor prioridad. Se activa cuando el agente detecta un obstáculo o a otro agente en la celda hacia la que planea moverse. Cuando sucede esto el agente ejecuta acciones como detenerse, rodear el obstáculo o rebasar en diagonal si la celda a la que va está libre, con esto se logran evitar la mayoría de los choques.

Capa 2: Reglas de tránsito

Cuando no hay riesgo de choque, toma control la capa encargada de asegurar que el agente respete las restricciones del entorno. En esta capa el agente sigue el sentido de las calles, se detiene ante un semáforo en rojo y evita cambios de dirección no permitidos, con esto evitamos que el agente tome decisiones peligrosas en el ambiente.

Capa 3: Seguimiento de ruta

El agente ejecuta su ruta calculada mediante búsqueda por anchura. Esta capa dirige el avance del agente sobre el gráfico de intersecciones, seleccionando la siguiente dirección hacia su destino. Aquí se ejecuta el movimiento y el avance en la ruta

Capa 4: Ajuste de ruta

Si el camino planificado se encuentra obstruido, esta capa activa mecanismos para seleccionar rutas alternativas dentro del grafo disponible. Es un comportamiento que permite adaptarse a las condiciones del tráfico.

Capa 5: Llegar al objetivo

Esta capa final asegura que el agente llegue al destino, y ejecuta movimientos para poder dirigirse y calcular su posición en base al destino para poder posicionarse sobre él, marcando que el agente ha terminado su ruta.

Características del ambiente:

- **Inaccesible:** Aunque los agentes conocen desde el inicio la estructura general del mapa así como los sentidos de las calles, la ubicación de los semáforos y de los obstáculos, no cuentan con información actualizada sobre la posición o trayectoria de los demás agentes. Esta falta de visibilidad en tiempo real implica que cada agente debe tomar decisiones basadas en información parcial e incompleta. Además, el hecho de que nuevos vehículos se incorporen cada 10 steps hace que el entorno siga siendo desconocido. Por estas razones, el ambiente no es accesible.
- **Determinístico:** El ambiente es determinístico porque las acciones de cada agente tienen resultados consistentes. Si un agente decide avanzar, girar o detenerse en respuesta a un semáforo o a un obstáculo, el resultado siempre será el mismo bajo las mismas condiciones. No existen factores aleatorios dentro de la simulación que modifiquen el comportamiento de los agentes o del entorno.
- **Episódico:** La simulación está estructurada de manera que cada episodio representa únicamente la situación actual del agente. Las decisiones del presente no dependen de episodios futuros porque los agentes no planean a largo plazo más allá de seguir su ruta hacia el destino. Además, los agentes no almacenan un historial de interacciones previas con el entorno ni con otros agentes. Esto hace que el ambiente sea episódico.
- **Estático:** El ambiente es estático porque no existen cambios que no estén en el control de los agentes. Los agentes modifican la dinámica del ambiente conforme avanzan, pero estos cambios solo dependen de las acciones de los propios agentes.
- **Discreto:** El entorno es discreto porque tanto los estados como las acciones de los agentes son claramente fijas/finitas. Las acciones posibles de los agentes son finitas y están delimitadas por la arquitectura de subsunción.

Conclusiones

En conclusión, uno de los mayores retos fue la implementación de un grafo para representar el mapa esto debido al cambio del mapa de este año, ya que nuestro algoritmo para conectar grafos estaba diseñado para ciertas características de los mapas pasados, como carriles de dos de anchura, y crear las rutas le costó al algoritmo en este nuevo mapa . En un principio pensamos

que la implementación de BFS como algoritmo de búsqueda era el más óptimo para permitir que los agentes encontraran rutas eficientes de manera sencilla y confiable, sin embargo encontramos que otros algoritmos como A* hubieran optimizado este proceso y permitido que más agentes llegaran a su destino. A pesar de esto, logramos una simulación que ofrece una visualización de movilidad urbana básica y que se podría ampliar fácilmente incorporando algoritmos más avanzados, comportamiento colaborativo o detección dinámica de congestiones, esto debido también a cómo se gestionan los nodos en el grafo, dado que al momento de recalcular la ruta por las condiciones de los nuevos mapas y los carriles nuevos de anchura de una celda, es muy complejo recalcularlo, haciendo que muchos agentes tengan asignadas rutas iguales, generando tráfico.

Video demo:

<https://drive.google.com/file/d/1xmZq6lwPJhsh2tIFTaUz-U7zZcO0XHYk/view?usp=sharing>