



Campus Santa Fe

Reporte del reto

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 301)

Héctor Julián Zárate Ramirez | A01027743

Fernando Adrián Fuentes Martínez | A01028796

Gilberto Echeverría Furió

Octavio Navarro Hinojosa

29 de Noviembre de 2024

Proceso de instalación:

1. Instalación de dependencias Flask y Mesa:

Moverse a la carpeta \ModelacionMultiagentes\Server, y correr:

```
pip install mesa==2.4.0 flask flask_cors
```

2. Iniciar lógica de Mesa con API:

```
python server.py
```

3. Visualización WebGL:

Moverse a la carpeta \ModelacionMultiagentes\WebGL, y correr:

```
npm install
```

4. Iniciar Front de WebGL:

```
npx vite
```

1. Problema y propuesta de solución

Problema:

El tráfico en las ciudades es uno de los mayores desafíos de la vida moderna, es complejo porque depende de muchos factores, como la cantidad de vehículos en circulación, el diseño de las carreteras, el funcionamiento de los semáforos y la aparición de imprevistos, como accidentes o bloqueos.

Objetivo:

La idea detrás de esta simulación es recrear cómo funciona el tráfico en una ciudad dentro de un entorno controlado. El propósito es encontrar formas de manejar el flujo vehicular de manera más eficiente, reduciendo las congestiones, logrando que los vehículos lleguen a su destino lo más rápido posible y analizando cómo diferentes configuraciones pueden influir en la movilidad general.

Propuesta de solución:

Para abordar este problema, proponemos crear una simulación donde diversos elementos interactúan entre sí dentro de una cuadrícula (grid). Los elementos principales incluyen:

- **Vehículos:** Se moverán por la cuadrícula siguiendo rutas óptimas hacia sus destinos, calculadas con el algoritmo A*.
- **Semáforos:** Regularán el paso de los vehículos alternando entre luz verde y roja según tiempos preestablecidos.
- **Destinos:** Serán los puntos a los que los vehículos intentarán llegar.

- **Carreteras:** Espacios por donde los vehículos pueden transitar.
- **Obstáculos:** Elementos que bloquean el paso (Edificios)

En esta simulación, cada elemento tendrá un conjunto de reglas que determinará su comportamiento. Por ejemplo, los vehículos seguirán el camino más corto hacia su destino, respetando los semáforos y evitando colisiones con otros vehículos u obstáculos. Los semáforos, por su parte, gestionarán el flujo vehicular en las intersecciones, permitiendo el paso alternado en diferentes direcciones.

2. Diseño de los agentes

Car

- **Objetivo:** Llegar a su destino recorriendo el camino más corto y evitando bloqueos.
- **Capacidad efectora:**
 - Movimiento dentro del grid siguiendo la ruta más eficiente.
 - Evita colisiones con otros vehículos y respeta el estado de los semáforos.
- **Percepción:**
 - Identifica carreteras, semáforos, obstáculos y su posición actual dentro del grid.
- **Proactividad:** Determina dinámicamente su ruta utilizando el algoritmo A*.
- **Métricas de desempeño:** Número de coches que alcanzan su destino.

Traffic_Light

- **Objetivo:** Regular el flujo vehicular alternando entre rojo y verde.
- **Capacidad efectora:** Cambia su estado cada cierto número de pasos.
- **Percepción:** No tiene percepción activa, pero afecta al flujo de los coches.
- **Proactividad:** Cambia su estado basado en el tiempo definido.
- **Métricas de desempeño:** No aplica.

Destination

- **Objetivo:** Actuar como punto final para los coches.
- **Capacidad efectora:** No aplica, ya que es un objetivo estático.
- **Percepción:** No tiene percepción ni interacción con otros agentes.
- **Proactividad:** No aplica.
- **Métricas de desempeño:** No aplica.

Obstacle

- **Objetivo:** Representar elementos que bloquean el paso.
- **Capacidad efectora:** No tiene capacidades efectoras, ya que es un objeto estático.
- **Percepción:** No tiene percepción ni interacción con otros agentes.
- **Proactividad:** No aplica.
- **Métricas de desempeño:** No aplica.

Road

- **Objetivo:** Determinar los caminos disponibles y sus direcciones.
- **Capacidad efectora:** Define la dirección permitida para los coches en cada celda.

- **Percepción:** No tiene percepción activa.
- **Proactividad:** No aplica.
- **Métricas de desempeño:** No aplica.

3. Arquitectura de subsunción de los agentes

Explorar:

Condición: Cuando no hay bloqueos ni semáforos en rojo en el camino.

Acción: Avanzar hacia el destino siguiendo la ruta más corta.

Esperar en semáforo:

Condición: Encontrar un semáforo en rojo.

Acción: Permanecer en la celda actual hasta que el semáforo cambie.

Evitar colisiones:

Condición: Detectar otro carro en la celda destino inmediata.

Acción: Reevaluar la ruta o esperar hasta que la celda esté libre..

Finalizar viaje:

Condición: Al alcanzar la celda destino o una celda vecina válida.

Acción: Detenerse, retirarse del grid y ser eliminado del modelo.

4. Características del ambiente

- **No determinístico:**
Las rutas de los coches se recalculan dinámicamente utilizando el algoritmo A*, además, los semáforos cambian de estado siguiendo intervalos predefinidos, lo que introduce incertidumbre en el flujo vehicular.
- **Accesible:**
El ambiente es parcialmente accesible porque los coches solo perciben celdas adyacentes en su entorno inmediato. Esto incluye información sobre carreteras, semáforos, obstáculos y otros coches dentro de esa vecindad limitada.
- **No episódico:**
Las acciones actuales de los agentes, como elegir una ruta o esperar en un semáforo, afectan estados futuros. Por ejemplo, tomar un desvío puede evitar tráfico más adelante, pero también podría retrasar la llegada al destino.
- **Dinámico:**
El ambiente cambia constantemente. Los coches se mueven hacia sus destinos, los semáforos alternan entre verde y rojo, y la presencia de coches en diferentes celdas puede generar bloqueos o afectar las rutas disponibles.
- **Discreto:**
El entorno está modelado como un grid de celdas, donde cada celda puede contener un agente

(carro, semáforo, destino, obstáculo o carretera) y los agentes solo se mueven entre celdas definidas.

5. Estadísticas recolectadas en las simulaciones

La simulación recolecta el número de coches que llegaron a su destino y el total de coches activos en el grid.

6. Conclusiones

La simulación de tráfico es una herramienta útil para entender cómo se comporta el tráfico en una ciudad y probar diferentes formas de mejorar la movilidad. Permite observar cómo interactúan los coches, semáforos, carreteras y obstáculos, y cómo estas interacciones afectan el flujo vehicular y la congestión en diferentes áreas.

Una de las fortalezas del modelo es que los coches pueden encontrar rutas eficientes hacia sus destinos utilizando el algoritmo A*. Esto podría reflejar el funcionamiento de sistemas de navegación modernos. Además, los semáforos ayudan a regular el tráfico y permiten probar distintos tiempos de cambio para encontrar configuraciones que mejoren la fluidez del tránsito aunque hay ciertos detalles que podrían volver más realista o eficiente la simulación, en este reto los vehículos solo ven las celdas que están a su alrededor inmediato. Si pudieran "ver" más lejos, podrían anticipar problemas y tomar mejores decisiones.

En general, esta simulación podría ser una buena herramienta para planificar y mejorar el transporte en las ciudades.

link del repo:

<https://github.com/HJZR2004/ModelacionMultiagentes.git>