Tecnológico de Monterrey

Campus Santa Fe

SIMULACIÓN DE MOVILIDAD URBANA EN UNA SIMULACIÓN COMPUTACIONAL

Curso:

Modelación de sistemas multiagentes con gráficas computacionales

Grupo:

302

Profesores:

Octavio Navarro Hinojosa

Gilberto Echeverría Furió

Presentan:

Gabriel Muñoz Luna - A01028774

Miguel Enrique Soria - A01028033

Fecha de entrega:

29 - 11 - 2024

1. Introducción

La movilidad urbana se define como la habilidad de transportarse de un lugar a otro y es fundamental para el desarrollo económico y social y la calidad de vida de los habitantes de una ciudad. Desde hace un tiempo, asociar la movilidad con el uso del automóvil ha sido un signo distintivo de progreso. Sin embargo, esta asociación ya no es posible hoy. El crecimiento y uso indiscriminado del automóvil —que fomenta políticas públicas erróneamente asociadas con la movilidad sostenible—genera efectos negativos enormes en los niveles económico, ambiental y social en México.

Durante las últimas décadas, ha existido una tendencia alarmante de un incremento en el uso de automóviles en México. Los Kilómetros-Auto Recorridos (VKT por sus siglas en Inglés) se han triplicado, de 106 millones en 1990, a 339 millones en 2010. Esto se correlaciona simultáneamente con un incremento en los impactos negativos asociados a los autos, como el smog, accidentes, enfermedades y congestión vehicular.

Para que México pueda estar entre las economías más grandes del mundo, es necesario mejorar la movilidad en sus ciudades, lo que es crítico para las actividades económicas y la calidad de vida de millones de personas.

En este reporte se presentarán las formas en las que se simuló el tráfico en un mapa de una ciudad con rotondas, semáforos, direcciones de calle, edificios y destinos para los coches, buscando un enfoque que reduzca la congestión vehicular dentro de la simulación.

2. Solución Propuesta

Como problema presentado en la introducción, el alto incremento y uso indiscriminado del automóvil genera efectos negativos en los niveles económicos, ambientales y sociales en México.

Se busca desarrollar una simulación de multiagentes de vehículos que transiten en una ciudad que simulan el tráfico generado en la ciudad de México, estos agentes se mueven hacia un destino respetando reglas vehiculares y evitando generar congestión vehicular. La solución empieza generando coches en cada esquina del mapa a los cuales se les asigna uno de los destinos existentes y estos tendrán que encontrar la forma más eficiente de llegar evitando generar tráfico, los vehículos conocen el mapa por lo que a través de un algoritmo de búsqueda en este caso BFS deben encontrar la ruta más óptima. El coche tiene la habilidad de respetar reglas y en caso de encontrar tráfico, poder moverse, buscar otra ruta, con la finalidad de encontrar la mejor opción de optimizar el tráfico generado. Como sé

mencionó respetan reglas, las cuales son principalmente no pasarse los semáforos en rojo, estos semáforos van alternando entre verde y rojo como en la vida real.

3. Diseño de la Solución

3.1. Simulación de tráfico

Utilizando el módulo Mesa de Python en su versión 2.4.0, se programó un modelo de mesa para simular una ciudad basada en un mapa leído a la par de un diccionario para representar correctamente los sentidos de las calles, los edificios, los semáforos, los destinos y los coches. Como parte de la funcionalidad de mesa, los componentes del entorno (semáforos, calles, edificios y destinos) se programaron con la clase *Agent* de mesa. Sin embargo, estos no son agentes y son parte del ambiente. Esto para que mesa pueda representarlos dentro de la simulación.

3.1.1 Diseño de los agentes

Contamos con un solo agente, el coche, el cual busca llegar a su destino de la forma más eficiente. El agente utiliza un algoritmo de búsqueda, en este caso es un BFS que a través de un grafo dirigido diseñado al momento de leer el mapa encuentra la ruta más óptima para llegar al destino, y al momento de realizar un cambio de carril, ya sea porque hay tráfico o porque está libre en diagonal, vuelve a calcular la mejor ruta. El agente cuenta con reactividad porque pueden reaccionar a los coches de enfrente, a los semáforos y al tráfico. Son proactivos porque buscan el camino más corto y toman decisiones en intersecciones.

PEAS del carro

Performance

El agente es capaz de navegar por el entorno siguiendo una ruta específica, puede llegar a un destino desde su posición inicial, evita en lo mayor de lo posible el tráfico, se adapta al encontrar tráfico.

Environment

El entorno al que se enfrenta el agente es una cuadrilla que representa una ciudad con distintos entes, entre ellos están los caminos los cuales tienen establecidas direcciones, obstáculos que son edificios y no se puede pasar por ellos, destinos que son lugares a los que tiene que llegar el coche, semáforos que cambian entre rojo y verde, este entorno va cambiando por el tráfico.

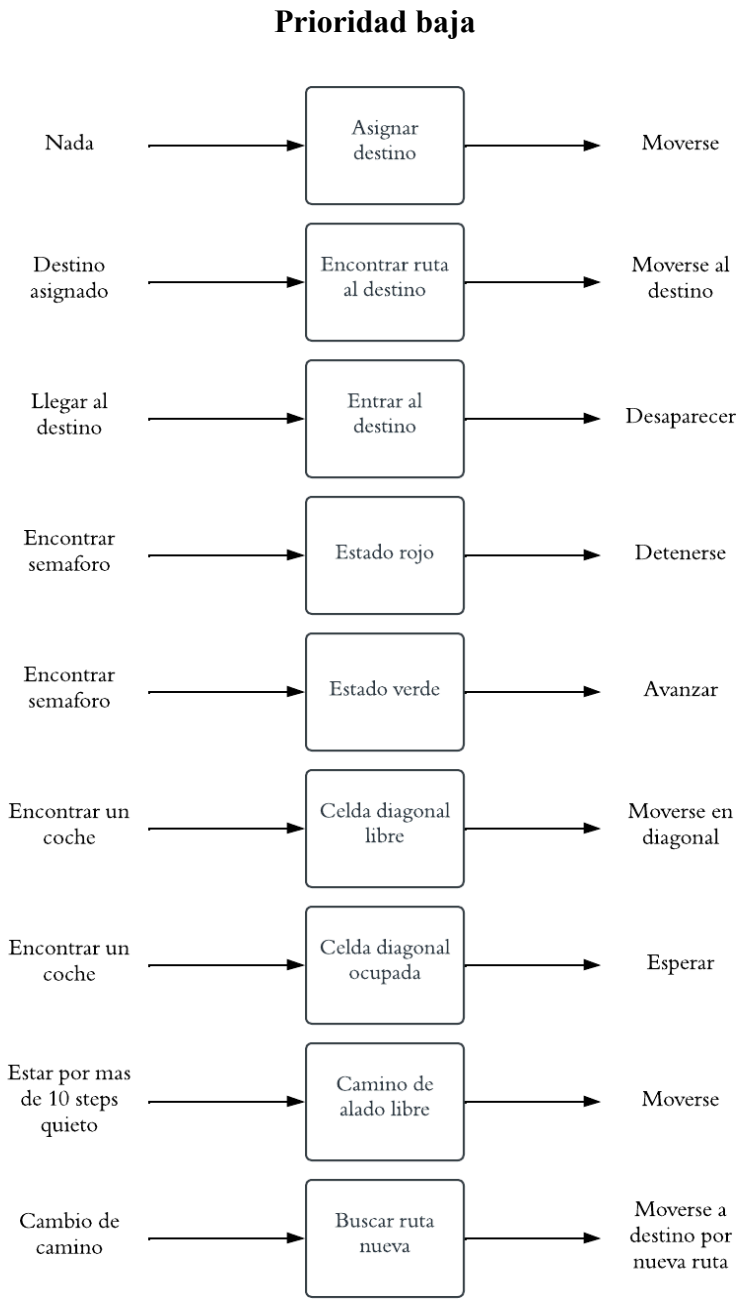
Actuators

El coche se puede mover en cuatro direcciones "Up", "Down", "Left", "Right", en las calles siguiendo la ruta.

Sensors

Identifica lo que le rodea, si es un semáforo, una calle, un obstáculo, un destino y en caso de encontrar un semáforo se puede detener, conoce las direcciones de las calles, detecta si ha estado parado mucho tiempo, y sabe cuál es su destino.

3.1.2 Arquitectura de subsunción



3.1.3 Características del ambiente

Nuestro ambiente, como ya se mencionó, es una simulación de una ciudad, en la que los coches transitan con la finalidad de llegar a su destino.

Es un ambiente medianamente accesible porque los agentes de coche conocen el mapa, junto con la posición de los obstáculos, semáforos, destinos y dirección de las calles, lo que les permite generar la ruta más óptima a su destino. Pero no conocen el estado en tiempo real de los semáforos, ni el tráfico existente.

En su mayoría es no determinista, ya que aunque los autos cuentan con acciones deterministas como empezar siempre en una esquina, o las direcciones de las cuales están establecidas, los lugares de destino siempre son aleatorios, no siempre es la misma cantidad de agentes que aparecen y a unque aparecen en las esquinas tienen oportunidad de realizar diferentes acciones como cambiar de carril o buscar una nueva ruta a su destino evitando en lo mayor de lo posible el tráfico.

Es no episódico, ya que los autos no dependen únicamente de su estado actual, al conocer la ruta que deben seguir “recuerdan” por donde han pasado y saben a dónde se deben mover, en caso de encontrar tráfico pueden desviarse de su ruta actual lo que les permite buscar otro camino a su destino evitando la ruta por la que iban.

Al ser un ambiente discreto en cuanto a tamaño es fijo, el entorno es una ciudad con límites y tanto los edificios, semáforos, destinos y lugares de aparición se encuentran establecidos, también las direcciones están definidas como “up”, “left”, “down”, “right”. Los semáforos solo cambian entre dos estados y existe un conjunto fijo de acciones a realizar. Por último, el modelo trabaja en pasos

Es un ambiente dinámico debido a que existen cambios a lo largo del mapa, entre ellos están los semáforos van cambiando entre rojo o verde, las calles pueden ser una u otra dirección y las acciones del coche se ven afectadas por si existe tráfico.

3.2 Simulación de WebGL

3.2.1 Herramientas utilizadas

Se utilizó un servidor de Vite con JavaScript para representar los coches en la ciudad utilizando el módulo de twgl en un entorno 3D simulado con WebGL2. También se utilizó lilgui para tener sliders con los que modificar parámetros dentro de la simulación, como la posición de la cámara y los valores, colores y posición de la luz. El resultado de esto es un render 3D de la simulación con modelos importados representando la simulación realizada en mesa.

3.2.2 Simulación

La simulación importa shaders programados con la lógica de simulación de luces Phong, tomando también un valor de atenuación en distancia utilizado para que las luces de los semáforos no generen una luz demasiado fuerte, lo que causaría que la escena se alumbre en amarillo al mezclar las luces de todos los semáforos. Para la fluidez de la simulación, se utilizó una variable que indica cada cuantos segundos se desea que se actualice la simulación. La simulación calcula una variable que mide el tiempo transcurrido entre frames, se calculan las posiciones intermedias de cada agente, dando como resultado una animación fluida.

4. Conclusiones

Se corrió la simulación con generación de 4 coches nuevos en las esquinas cada 10, 5 y 2 steps y se vieron los siguientes resultados:

1.

Cada 10 pasos, se genera un nuevo agente en las 4 esquinas

Simulación 1 a 50 pasos totales

```
Paso 50: Agentes actuales = 20, Agentes que llegaron a su destino = 4, Paso actual: 50
```

Simulación 1 a 750 pasos totales

```
Paso 575: Agentes actuales = 22, Agentes que llegaron a su destino = 210, Paso actual: 575
```

Simulación 2 a 50 pasos totales

```
Paso 50: Agentes actuales = 18, Agentes que llegaron a su destino = 6, Paso actual: 50
```

Simulación 2 a 750 pasos totales

```
Paso 575: Agentes actuales = 17, Agentes que llegaron a su destino = 215, Paso actual: 575
```

2.

Cada 5 pasos, se genera un nuevo agente en las 4 esquinas

Simulación 1 a 50 pasos totales

```
Paso 50: Agentes actuales = 31, Agentes que llegaron a su destino = 13, Paso actual: 50
```

Simulación 1 a 750 pasos totales

```
Paso 575: Agentes actuales = 43, Agentes que llegaron a su destino = 414, Paso actual: 575
```

Simulación 2 a 50 pasos totales

```
Paso 50: Agentes actuales = 36, Agentes que llegaron a su destino = 7, Paso actual: 50
```

Simulación 2 a 750 pasos totales

```
Paso 575: Agentes actuales = 34, Agentes que llegaron a su destino = 425, Paso actual: 575
```

3.

Cada 2 pasos, se genera un nuevo agente en las 4 esquinas

Simulación 1 a 50 pasos totales

```
Paso 50: Agentes actuales = 77, Agentes que llegaron a su destino = 24, Paso actual: 50
```

Simulación 1 a 575 pasos totales

```
Paso 575: Agentes actuales = 116, Agentes que llegaron a su destino = 821, Paso actual: 575
```

Simulación 2 a 50 pasos totales

```
Paso 50: Agentes actuales = 85, Agentes que llegaron a su destino = 18, Paso actual: 50
```

Simulación 2 a 575 pasos totales

```
Paso 575: Agentes actuales = 120, Agentes que llegaron a su destino = 836, Paso actual: 575
```

A Través de las simulaciones realizadas tenemos que en promedio por cada 4 agentes llegan 3 a su destino, lo que se puede interpretar para que haya una mejoría en el tráfico, se tienen que respetar las reglas de vialidad y se mantienen en el carril más cercano a su destino.

En resumen, se logró implementar una simulación en mesa visualizada en WebGL donde los multiagentes programados se comportan de manera correcta, navegando la ciudad, respetando las reglas de tránsito con una ruta calculada desde su posición inicial a su destino asignado, evitando generar congestiones mediante cambios de carril y recálculo de rutas, demostrando que sí es posible tener un volumen alto de coches en una ciudad sin generar tráfico y que se estanquen todos los caminos, siempre y cuando existan rutas que lo permitan y los vehículos sigan la mejor ruta posible siguiendo las reglas de vialidad.