Reto: Movilidad Urbana

TC2008B.302: Modelación de sistemas multiagentes con gráficas computacionales

Sergio Zuckermann Gittler A01024831

Santiago Benitez A01782813

Bajo la instrucción de

Octavio Navarro Hinojosa

Gilberto Echeverría

30 de Noviembre de 2023

Introducción:

Problema:

La movilidad urbana, se define como la habilidad de transportarse de un lugar a otro1 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. Ésto 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.

Este reto propone una solución al problema de movilidad urbana en México, mediante un enfoque que reduzca la congestión vehicular al simular de manera gráfica el tráfico, representando la salida de un sistema multi agentes.

Utilizando el motor Unity para representar en tres dimensiones el modelo, manejar los objetos y las simulaciones mediante Scripts. Se utilizaron scripts para controlar a los agentes activos mediante manipulación de componentes como Meshes para representar movimiento y cambiar color de objetos para representar información del sistema. Se aplicaron conceptos como matrices de transformación para coordinar los diferentes objetos dentro de un plano, haciendo así que la visualización tuviera sentido y fuera acorde a un escenario real. Finalmente se crearon modelos, texturas y materiales para que los agentes fueran llamativos a la vista y representarán a sus contrapartes de la vida real.

También se usó el módulo Mesa para Python, para generar la logica de un modelo multiagentes usando una definición de agentes donde se refleja el diseño de estos de manera práctica, tomando en cuenta su toma de decisiones según sus sensores y finalmente la aplicación de la acción necesaria. Dentro de Mesa se usará una definición del modelo con

distintos parametros para definirlo, además que leera un conjunto de caracteres que representarán a los distintos agentes y tendrá control de atributos globales como la cantidad de pasos para la generación inicial y continua de distintos agentes. Finalmente se usará un servidor de Flask para establecer los APIs que serán los encargados de hacer la conexión con Unity y los endpoints para poder representar los cambios de Mesa en Unity.

En este documento se establece el diseño de los agentes para alinear los requerimientos necesarios para la simulación así como una descripción ambiental para incorporar a los distintos comportamientos de los agentes. Así como la arquitectura de subsunción que se usará para desarrollar el agente en Mesa. Después se hará una breve conclusion donde se expindran los puntos a mejorar del modelo, así como las virtudes y errores que se cometieron durante el desarrollo.

Desarrollo:

Agentes:

A.- Coche:

Objetivo: Llegar desde su origen hasta un destino establecido en la menor cantidad de tiempo, sin chocar con otros coches y respetando el estado de los semaforos.

Capacidad efectora:

El coche tendrá distintas acciones en su repertorio y será el único agente capaz de tomar decisiones conforme al estado del ambiente. Las acciones disponibles son: Calcular ruta, hacer desvio. pasar y moverse.

Percepción: El agente Coche puede escanear sus vecinos en dirección superior, inferior y lateral solo si los vecinos están a una casilla de distancia del agente.

PEAS:

- Performance
  - El agente es capaz de efectuar decisiones para resolver de mejor manera el problema teniendo en cuenta su camino actual y sus vecinos. Las acciones están jerarquizadas para tener mejores resultados.
- Environment
  - El ambiente contiene semaforos y obstaculos, además de calles direccionadas. Los semaforos determinan si el agente puede avanzar en ese turno (verde) o tendrá que pasar (rojo). Los obstaculos son estaticos y sirven más bien para delinear el mapa y por ende el camino más optimo para el agente ya que se deben esquivar a toda costa.

● Actuators

○ Los actuadores que el agente tiene son los que se describieron en su capacidad efecturadora y que serán expandidos en la arquitectura de subsunción.

● Sensors

○La percepción del agente está limitada a reconocer información solo de casillas aledañas (1 celda de distancia, máximo $\sqrt{2}$ en distancia euclidiana).

Proactividad:

El agente muestra una proactividad limitada, donde podrá moverse siempre y cuando los espacios siguientes de su ruta estén vacíos. Sin embargo, el agente pierde la proactividad cuando existe un flujo de agentes grande, al dejar de usar el camino más optimo, llegando hasta casos donde ya no busca continuar con su movimiento y queda bloqueado.

Reactividad:

Los agentes reaccionan al ambiente primariamente cuando encuentran otros agentes. Los semaforos por ejemplo, estando rojos haran reaccionar al agente de tal manera en que deje de seguir su ruta y se quede en el mismo lugar. Un agente coche siempre reacciona ante otros coches por su constante intención de evitar chocar lo cual hará que reaccionen cambiando su ruta, tomando un desvío o quedándose parados.

Métricas de desempeño:

Basicamente existe solo una metrica de desempeño que depende de tres factores. El mapa, el tiempo máximo (pasos) y la frecuencia de creación de coches. Entonces la metrica de desempeño sería:

Bajo un mapa constante y una frecuencia de creación establecida, cuantos coches pueden llegar a su destino antes de que se acabe el teimpo

B.- Obstaculo y destino (Agentes ambientales):

Objetivo: No tienen un objetivo.

Capacidad efectora: Son estaticos

Percepción: No tienen percepción

Proactividad: No son agentes proactivos

Métricas de desempeño: No tienen metricas de desempeño.

C.- Semaforo:

Objetivo: No tienen un objetivo.

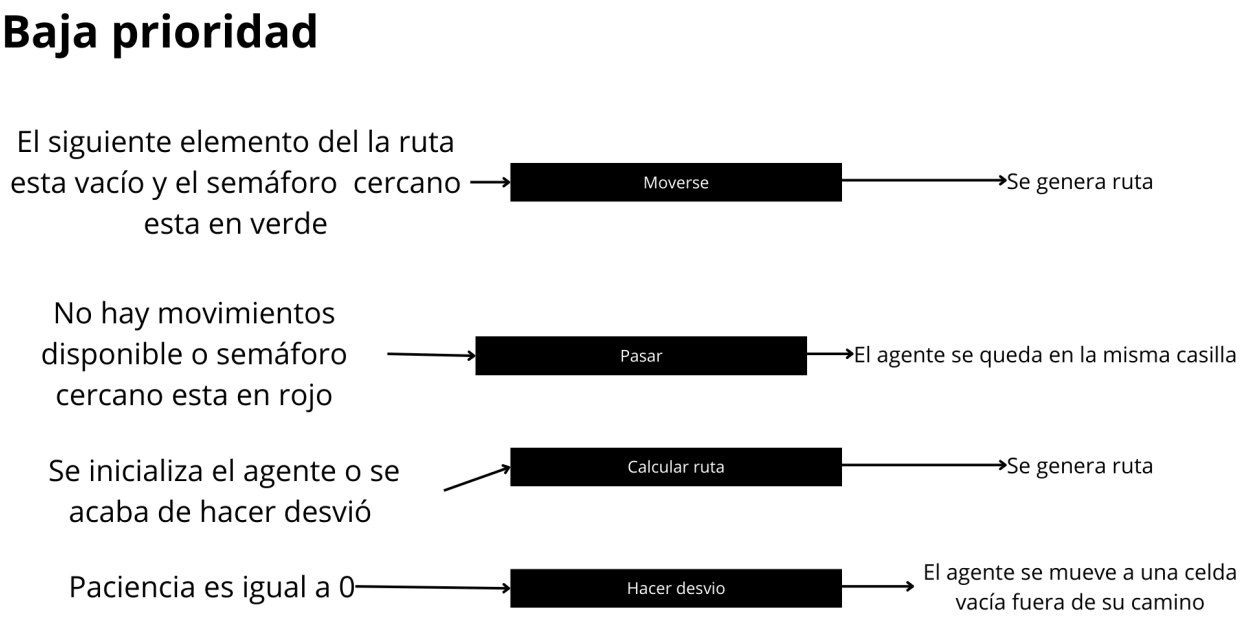
Capacidad efectora: Cambiar de estado de verde a rojo.

Percepción: No tienen percepción

Proactividad: No son agentes proactivos

Métricas de desempeño: No tienen metricas de desempeño.

Arquitectura de subsunción del agente:



Características del ambiente:

El ambiente será medianamente accesible pues el agente puede obtener información completa, precisa y actualizada sobre el estado del mismo solo si esa parte del ambiente está dentro de la percepción del agente.

El ambiente será determinista: Al no usar conceptos complicados como fisicas, no existen una cantidad enorme de variables. Basicamente el agente podrá tener acciones claras después de escanear su entorno, ya que el agente solo depende del ambiente en cuanto a donde están posicionados los obstaculos. No siempre tomará las mismas decisiones porque también depende del estado de los semaforos, pero tomará un curso de acción bastante parecido en todas las iteraciones. Lo que puede causar más incertidumbre es la posibilidad de encontrar otros agentes Coche, sin embargo estos no son parte del ambiente.

El ambiente será no episodico pues el agente sólo tomará decisiones basadas en sus percepciónes y estado actual sin tener en cuenta repercusiones en el futuro, la única politica episodica que llega a tomar el agente es el llegar a su casilla respetando el estado de los semaforos y las futuras posiciones de los vecinos Coche.

El ambiente será estatico pues se puede suponer que permanece sin cambios excepto por la realización de acciones por parte del agente. No existen otros procesos y cambios de maneras que salen del control del agente por lo que no es mayoritariamente dinamico, el único cambio sería el estado del semaforo.

Con todos estos atributos y parametros se puede deducir que aunque tenga incontables escenarios, estos no serán infinitos. Esto también incluye el hecho de que la simulación vive en un plano en dos dimensiones donde la mayoría de las cuadrículas ya están predefinidas si el mapa es constante, Existen un número fijo y finito de diferentes cuadrículas y escenarios por lo que el ambiente sería discreto.

Conclusiones.

La implementación de los agentes fue parcialmente exitosa, se pudó realizar correctamente la simulación dentro de Unity incorporando API y Endpoints de Flask. Sin embargo, el agente coche es muy inconsistente y hasta disfuncional cuando los parametros de creación de coches son muy altos, aunque cabe recalcar que bajo parámetros de creación más reducidos, la simulación trabaja de manera correcta. Esto se debe a una proactividad reducida, ya que el agente no recalcula las rutas en los momentos adecuados y solo toma la mejor ruta usando A-Star sin tomar en cuenta el flujo de otros agentes Coches. Esto lleva a una congestión de tráfico considerable que poco a poco se vuelve más grande y termina inmovilizado a todos los agentes. En futuras iteraciones de este modelo se recomienda rediseñar el cálculo de ruta para incorporar una busqueda que tome en cuenta casillas más lejanas y tenga en cuenta los futuros movimientos de los otros agentes. Una caracteristica que se deseaba agregar, pero por falta de tiempo fue descartada es tener una interacción social con los agentes donde se hagan solicitudes para que un coche deje pasar al otro y no “empaten”, perdiendo su proactividad. La peor limitante en el desarrollo de este modelo fue el tiempo, bajo presión solo se pudo agregar lo mínimo para que el agente trabaje bajo circunstancias ambientales. Otro factor que afectó a la construcción de la simulación fue la poca experiencia con el motor Unity en 3D, por lo que la realización de cambios simples tardó mucho más de lo esperado, retrasando asi el tiempo de desarrollo y por ende empujando a limitar las caracteristicas diferenciadoras del modelo. En el momento de realización de este reporte se está tratando de rediseñar el agente,

sin embargo por falta de tiempo no se podrán ver reflejados los cambios finales en este escrito, así que se describe la idea general: Recalcular rutas y aumentar la proactividad del agente cuando se vea enfrentado con más coches. Como cualquier modelo de agentes reactivos, solo se podrá avanzar haciendo un ciclo de diseño implementación, análisis, conclusiones constante.