



# Tecnológico de Monterrey

## **Evidencia 2. Avances y presentación del reto**

Gerardo Gutiérrez Paniagua, A01029422

Mateo Herrera Lavalle A01751912

Francisco Daniel Salcedo Catalán A01633010

Regina Rodríguez Sánchez A01284329

Modelación de sistemas multiagentes con gráficas computacionales

Prof. Gilberto Echeverría

Prof. Octavio Navarro

28 de noviembre del 2022

## Introducción

En este proyecto se busca simular el movimiento de los automóviles en las calles de una ciudad. Esto con el propósito de analizar las diferentes interacciones que suceden cuando varios automóviles intentan llegar a diferentes puntos en algún lugar, al igual que el tráfico que podría verse generado.

Para esto, se hará uso de un sistema multiagentes en el cual agentes que fungen como automóviles deberán moverse en la ciudad para llegar a diferentes destinos.

## Diagrama de Clases

Dentro de este sistema multiagentes se pueden encontrar los agentes señalados en la figura 1, la cual representa un diagrama de clases de estos mismos.

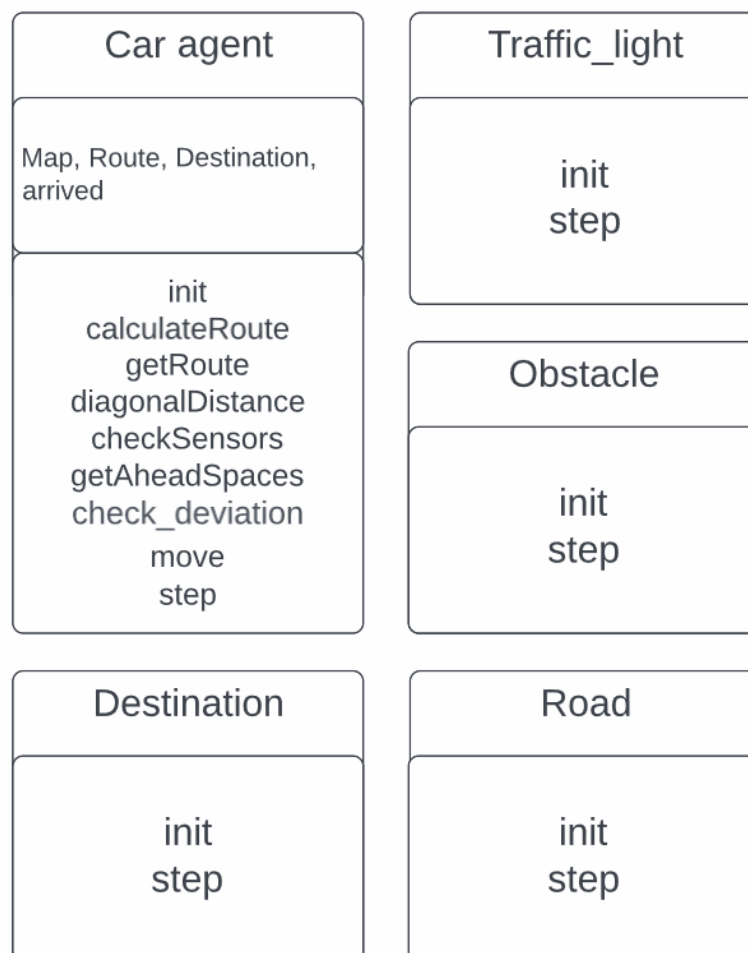


Figura 1: Diagrama de clases

## Diagrama de Prioridades

Una representación de la toma de decisiones del Agente Auto se puede observar en la siguiente figura.

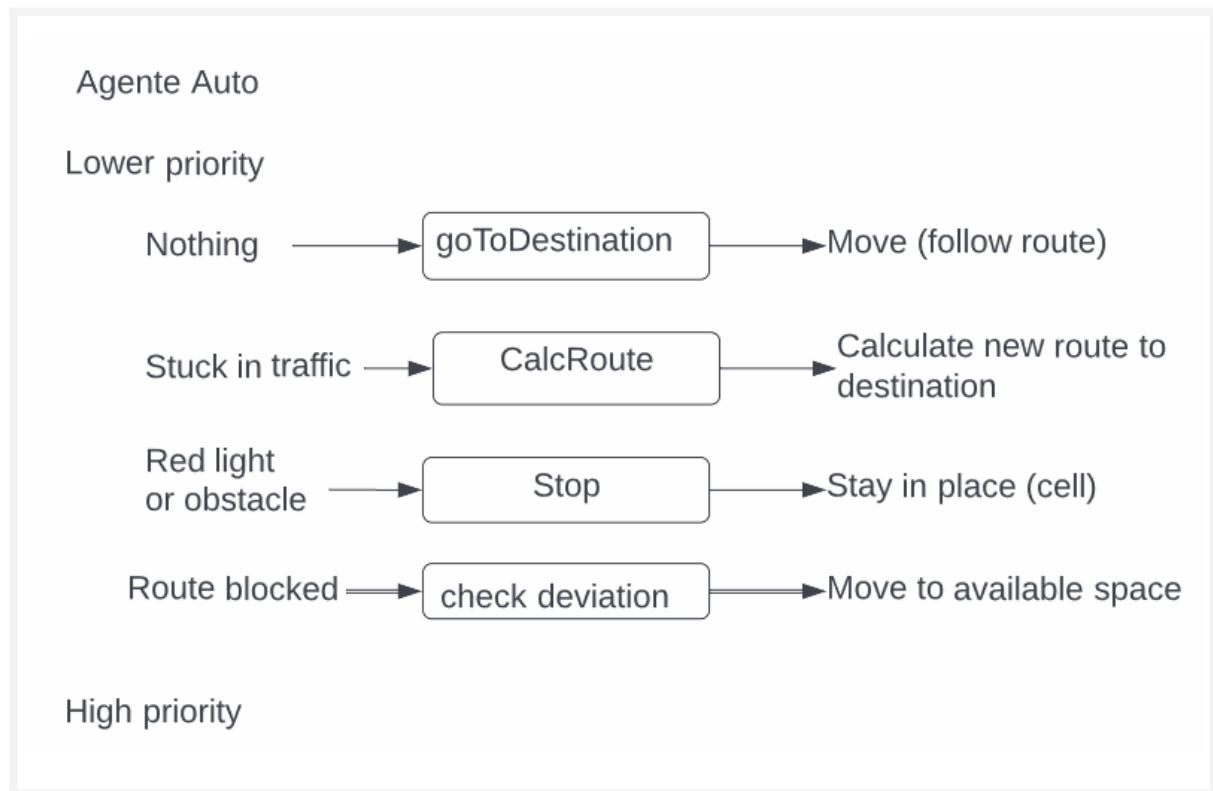


Figura 2: Diagrama de prioridades

# Pasos de Ejecución

## Pasos para ejecutar simulación en Mes

1. Tener acceso a una herramienta Git.
2. Clonar el repositorio localmente.  
`git clone https://github.com/G3RZZZ/TC2008B`
3. Abrir un ambiente que contenga python 3.8, Flask, y Mesa  
Usando Anaconda:
  - `conda create <nombre>`
  - `conda install python=3.8`
  - `pip install mesa`
  - `pip instal flask`
4. Ejecutar el servidor en la dirección  
'...\TC2008B\TrafficSimulation\MesaVisualization\TrafficSimulation' con el comando 'python server.py'
5. Inicializar la simulación con el botón 'Start'

## Pasos para ejecutar simulación en Unity

1. Tener acceso a una herramienta Git.
2. Clonar el repositorio localmente.  
`git clone https://github.com/G3RZZZ/TC2008B`
3. Abrir un ambiente que contenga python 3.8 y Flask  
Usando Anaconda:
  - a. `conda create <nombre>`
  - b. `conda install python=3.8`
  - c. `pip instal flask`
4. Ejecutar el servidor en la dirección '...\TC2008B\TrafficSimulation\Server' con el comando python server.py
5. Abrir el proyecto 'UnityVisualization' en Unity  
( '...\TC2008B\TrafficSimulation\UnityVisualization' )
6. Con el proyecto abierto, abrir la escena BuildCity
7. Inicializar simulación con el botón Play.

Enlace a repositorio: <https://github.com/G3RZZZ/TC2008B>

## Conclusiones individuales

Mateo Herrera Lavalle

A lo largo de estas cinco semanas, se simuló el tráfico de una ciudad mediante el uso de un sistema multiagentes en donde ‘agentes carro’ se desplazarán para llegar a las coordenadas de ‘agentes destino’. Para esto, nos centramos en modelar a nuestros agentes de tal manera que tuvieran la habilidad de reaccionar a cambios en su ambiente, manteniendo también la idea de que estos agentes lleguen a su destino con la ruta más óptima posible. Además, queríamos que estos agentes se comporten de manera realista, por lo que buscamos que estos respetaran leyes de tránsito y tuvieran cierto grado de comportamiento individual. Es por esto último por lo que decidimos que el modelo funcionaría con Random Activation, permitiendo así que no todos los agentes tomarán decisiones al mismo tiempo.

En cuanto a los agentes, decidimos que las variables que estos deberían tomar en consideración fueran de aspectos generales de la simulación a situaciones ya un poco más específicas. En general, los agentes siempre evalúan que el espacio al que se van a mover no contenga otro agente u obstáculo, además de revisar que los semáforos tengan luz verde cuando manejen a través de ellos. De igual manera, se permitió que los agentes se desviarán un poco de la ruta, para así permitir que estos cambiaran de carril y buscarán alternativas cuando existiera tráfico en frente de ellos. En situaciones más particulares, los agentes también revisan si tienen otro agente carro en su espejo retrovisor, para así evitar que estos se cruzaran e intersectan entre sí.

Estas condiciones lograron cumplir los objetivos propuestos en la simulación, ya que al introducirse un gran número de agentes de manera constante, esta pudo seguir funcionando sin un embotellamiento total por mucho tiempo. De hecho, después de correr la simulación por un largo periodo de tiempo, se pudo encontrar que de 370 agentes posibles, la simulación logró seguir funcionando hasta que se alcanzó el número de 361 agentes (Figura 1) activos, batiendo el récord de agentes establecido en años anteriores.

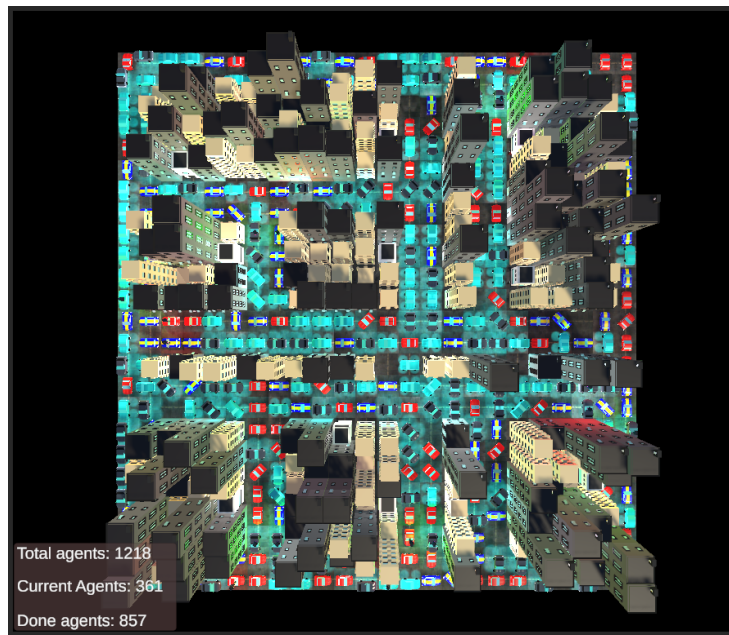


Figura 3: Embotellamiento total de agentes

El diseño gráfico de nuestros agentes surgió de ideas acerca de una ciudad futurista, en donde los carros vuelan y todo tiene un cierto aspecto distópico. Es decir, que esta ciudad es una en donde a pesar de tener avances tecnológicos, no pudieron solucionar problemas ambientales por lo que todo es un poco más gris. También nos gustó la idea de jugar con las luces que los coches producían, por lo que la ciudad se alumbra más en relación a cuántos coches hay dentro de la simulación

Considero que la mayor ventaja de la solución que fue implementada es justamente la flexibilidad y reactividad de nuestros agentes. Esto fue lo que permitió que la simulación corriera por mucho tiempo, además de evitar que se detuviera debido a un gran número de cálculos del algoritmo A\*. Las desventajas surgen de que los agentes en realidad no se comunican, lo cual causa que tráfico se acumule de manera mas rapida. Considero que implementar un cierto grado de comunicación entre agentes en donde se genere algo parecido a luces de cambio de dirección facilita estas interacciones. Además, creo que al hacer esto, generamos comportamientos más interesantes entre los agentes de la simulación.

Finalmente, considero que en realidad si obtuve muchos nuevos conocimientos en la realización de este proyecto. Nunca había usado blender, por lo que aprender un poco sobre esta herramienta y sus aplicaciones me pareció muy interesante. Además, dado que trabajé bastante en la implementación de los agentes, en realidad logré adquirir un incierto tipo de razonamiento específico que es necesario cuando se implementan comportamientos en este tipo de sistemas. Considero que mis expectativas de este curso si se cumplieron, y simplemente me hubiera gustado un poco más que se tratara de un bloque más largo.

## Gerardo Gutierrez Paniagua

La perspectiva de la implementación del modelo multiagentes fue priorizando que el agente respondiera a sus alrededores con activación aleatoria, conociendo solamente su destino y el mapa, debía navegar las y reaccionar a las circunstancias que se le presentarán. De manera que, los agentes interactúan con ellos mismos, bloqueando caminos y generando tráfico entre ellos. Con el fin de mantener el mismo criterio en los agentes para evaluar un camino libre de tráfico, se colocó un tiempo de reacción equivalente a dos steps; esto lo interpretamos como la paciencia del agente en reaccionar al ambiente.

En las implementaciones finales se tomó la decisión de priorizar el movimiento a espacios libres sobre mantenerse estrictamente en la ruta calculada, lo cual permitió que más agentes pudieran entrar a la simulación y también que se viera una representación realística del comportamiento de conductores humanos.

El contexto del diseño gráfico presentado viene a partir de imágenes y videos que se refieren a un futuro distópico, donde los autos vuelan en una atmósfera pesada donde la luz es artificial y el paisaje es denso en edificios monótonos. Para los coches, se usaron modelos con llantas y con cohetes para pretender su levitación, existen dos tipos de modelos para cada versión.

Como fue mencionado anteriormente, la implementación de un timer a los agentes, al igual que la libertad de desviarse ligeramente de la ruta calculada con el fin avanzar, son grandes ventajas, no solo por asemejar el comportamiento humano hasta cierto punto, pero también porque permitieron que más agentes Auto existieran dentro de la simulación, aprovechando al máximo los espacios del mapa, encontramos que este tipo de reacciones e interacciones puede ser catalogado como comportamiento emergente.

La desventaja principal en la solución es la falta de comunicación entre agentes, ya que como se había observado en pruebas anteriores, el comportamiento puede beneficiarse enormemente de la sincronización de información en la simulación. Compartir rutas, posiciones, Destinos y estados actuales de semáforos podría hacer que la representación se vea un tanto artificial por la toma de decisiones y reacciones omniscientes que tendrían los agentes Auto al navegar el mapa.

Si se quisiera cambiar la falta de comunicación mencionada, se podría implementar algún sistema de listas o diccionarios con información de las celdas alrededor del mapa, de manera que, si dos agentes están lado a lado, estos podrían compartir el estado de sus celdas adyacentes, lo cual podría mejorar el cálculo de nuevas rutas o de la reacción a avenidas congestionadas en el mapa.

A lo largo de estas semanas pude aprender acerca de los diferentes tipos de agentes y modelos en sistemas multiagentes, estas bases fueron las necesarias para poder plantear una estrategia de implementación de las actividades integradoras. Decidiendo si los agentes se beneficiarán de calcular paso por paso o varios pasos tratando de anticipar situaciones; finalmente, los fundamentos utilizados para la actividad integradora 1 fueron pilares para diseñar el comportamiento de los agentes en la simulación de tráfico.

Me pareció muy interesante cómo al estar resolviendo bugs del agente auto, su comportamiento no solo se hacía más “normal” a un auto, sino que también se construye una personalidad al agente, algo a lo que en el equipo terminamos bautizado como paciencia, y que después definimos como comportamiento emergente.

Estar cambiando la jerarquía de las decisiones del agente era lo que terminaba por hacer la simulación más fluida, en vez de encontrar rápidamente calles llenas de tráfico. En el desarrollo del proyecto observe cómo priorizar la ruta a seguir, o el movimiento hacia adelante impactando enormemente al modelo.

Para este proyecto se limitó el comportamiento que podía tener el agente en ciertas situaciones con el fin de que la estética del comportamiento fuera “normal”, sin embargo, sería interesante y enriquecedor para entender el tema implementar soluciones un poco más complejas que programar ifs. Implementando un poco de comunicación entre agentes podría cambiar sustancialmente el modelo, obteniendo un resultado distinto a lo que es ahora, pero con la libertad de añadir otro tipo de comportamientos basado en la activación simultánea.

Regina Rodríguez Sánchez

Francisco Daniel Salcedo Catalán



## Plan de Trabajo

Para la correcta realización de este proyecto se ha estado siguiendo el plan de trabajo mostrado a continuación. Los objetivos y tareas que han sido completadas se muestran en las siguientes tablas, al igual que las tareas que aún siguen en progreso.

Actividad	Fecha Estimada	Fecha de realización	Responsable	Estatus	Esfuerzo(hrs )
-----------	----------------	----------------------	-------------	---------	----------------

Revisión 1					
Creación de repositorio	3/11/2022	3/11/2022	Gerardo	Completado	1 hrs
Creación de proyecto de unity	3/11/2022	3/11/2022	Gerardo	Completado	1 hrs
Definición del estilo del proyecto	14/11/2022	10/11/2022	Todos	Completado	1 hrs
Creación de autos en blender	18/11/2022	17/11/2022	Todos	Completado	6 hrs por persona

Revisión 2					
Creación de escena y materiales de prueba	18/11/2022	18/11/2022	Gerardo	Completado	5 hrs
Definición de comportamientos de los agentes	23/11/2022	23/11/2022	Mateo	Completado	1 hrs
Descripción de los agentes	23/11/2022	23/11/2022	Mateo	Completado	1 hrs
Creación de Diagramas	23/11/2022	23/11/2022	Gerardo	Completado	1 hrs

de Clases					
-----------	--	--	--	--	--

Revisión 3					
Implementación de algoritmo de búsqueda A*	24/11/2022	24/11/2022	Mateo	Completado	5 hrs
Implementación de agentes en mesa	25/11/2022	24/11/2022	Mateo	Completado	5 hrs
Implementación del modelo en mesa	26/11/2022	26/11/2022	Mateo	Completado	5 hrs
Pruebas del funcionamiento completo en mesa	28/11/2022	28/11/2022	Mateo/Gerardo	Completado	5 hrs
Bug Fixes del funcionamiento en mesa	28/11/2022	28/11/2022	Mateo/Gerardo	Completado	2 hrs
Creación de edificios en blender	29/11/2022	29/11/2022	Regina / Francisco	Completado	10 hrs
Creación de calles en blender	29/11/2022	29/11/2022	Regina / Francisco	Completado	8 hrs
Creación de semáforos en blender	29/11/2022	29/11/2022	Regina / Francisco	Completado	8 hrs

Revisión final					
Implementación de agentes en unity	29/11/2022	30/11/2022	Mateo	Completado	3 hrs

Implementación del modelo en unity	29/11/2022	30/11/2022	Mateo	Completado	3 hrs
Bug Fixes del funcionamiento en unity	29/11/2022	30/11/2022	Mateo/Gerardo	Completado	3 hrs