



MATERIA:

Modelación de sistemas multiagentes con gráficas computacionales

GRUPO:

101 - Semestre FJ-2023

PROFESORES:

Daniel López Salas

Jorge Mario Cruz Duarte

TAREA / PROYECTO:

MAS y CG: Reto: Entrega de Anteproyecto/Avance

EQUIPO 3

LISTA DE ESTUDIANTES :

A00830475 - Ángel Rigoberto García García

A01246337 - Bryan Manuel De la O Perea

A01197991 - Hiram Maximiliano Muñoz Ramirez

A01197655 - Raúl Isaí Murillo Alemán

FECHA: Viernes 17 de Marzo del 2023

FORMATO REPORTE TÉCNICO - RETO - TC2008B

1. INTRODUCCIÓN

La población de la ciudad de Monterrey ha crecido incontrolablemente durante las últimas décadas. Este incremento ha tenido un gran impacto sobre las vidas diarias de los ciudadanos. Los recursos e infraestructura de la ciudad no han crecido a la misma tasa de la población. Uno de los principales ejemplos de esta diferencia es el tráfico en la ciudad. Durante las “horas pico”, uno puede durar más de una hora dentro del tráfico. Resolver este problema implicaría un gran proyecto de ampliación de la infraestructura vial, algo que es costoso y tardado. Esto es un gran limitante para cualquier iniciativa, ya que se requieren hacer costosos estudios para corroborar la factibilidad de un proyecto. Se necesita una manera más sencilla de llevar a cabo estos estudios.

2. CRÉDITOS

Ángel Rigoberto García García - Modelador

Bryan Manuel De la O Perea - Modelador

Hiram Maximiliano Muñoz Ramirez - Programador

Raúl Isaí Murillo Alemán - Programador / Modelador

3. CONTEXTO Y PROBLEMA

Se necesita una manera de analizar intersecciones de la vida real, así como probar diferentes soluciones sin tener que hacer modificaciones a la intersección real. Para esto, se busca generar un modelo multiagente que simule el comportamiento del tráfico vehicular. Este modelo debe de ser capaz de simular cualquier tipo de intersección vehicular, sin importar su forma, composición, personalidad de los conductores, etc. Por lo tanto, debe de tener parámetros que adapten la simulación a cualquier situación real.

4. OBJETIVOS GENERALES

- Crear un modelo dinámico que permita simular cualquier tipo de tráfico vehicular.
- Calcular métricas de flujo de tráfico y mostrarlas al usuario.
- Poder generar redes vehiculares a partir de planos reales.

5. RESTRICCIONES

Crear un modelo que pueda simular el comportamiento de agentes vehiculares es una tarea muy compleja. Un vehículo analiza muchos aspectos de sus alrededores para decidir cómo comportarse. Para reducir la complejidad de la solución, se debe definir una lista limitada de aspectos que los vehículos usarán para tomar decisiones. En el caso de este modelo, el modelo de aceleración seleccionado únicamente tomará en cuenta el vehículo que está directamente enfrente de él. Igualmente, para realizar cambios de carril, los vehículos únicamente consultan el estado de sus vecinos inmediatos.

Respecto al modelado del entorno, hay una infinidad de configuraciones posibles. En este proyecto nos enfocamos específicamente en el intercambiador Constitución-Gonzalitos, para así evitar tener que lidiar con algunos casos extremos.

6. HISTORIAS DE USUARIO

- Como diseñador de redes vehiculares, debo poder observar el comportamiento de los vehículos para que pueda identificar áreas de oportunidad en el diseño actual.
- Como diseñador de redes vehiculares, debe poder modificar los parámetros del sistema para así analizar diferentes situaciones de un sistema vehicular.
- Como ingeniero civil, debo poder ver el diseño de la red vehicular para así poder aplicarlo a la vida real.
- Como socio formador, debe poder ver el efecto que cierta modificación tiene sobre la red vehicular para así poder mostrarlo al público.

7. DESCRIPCIÓN DEL SISTEMA MULTIAGENTE

El sistema consta de una cantidad N de agentes vehiculares circulando por un sistema vehicular. Los agentes tienen como objetivo moverse de una entrada del sistema hacia una salida, elegida aleatoriamente.

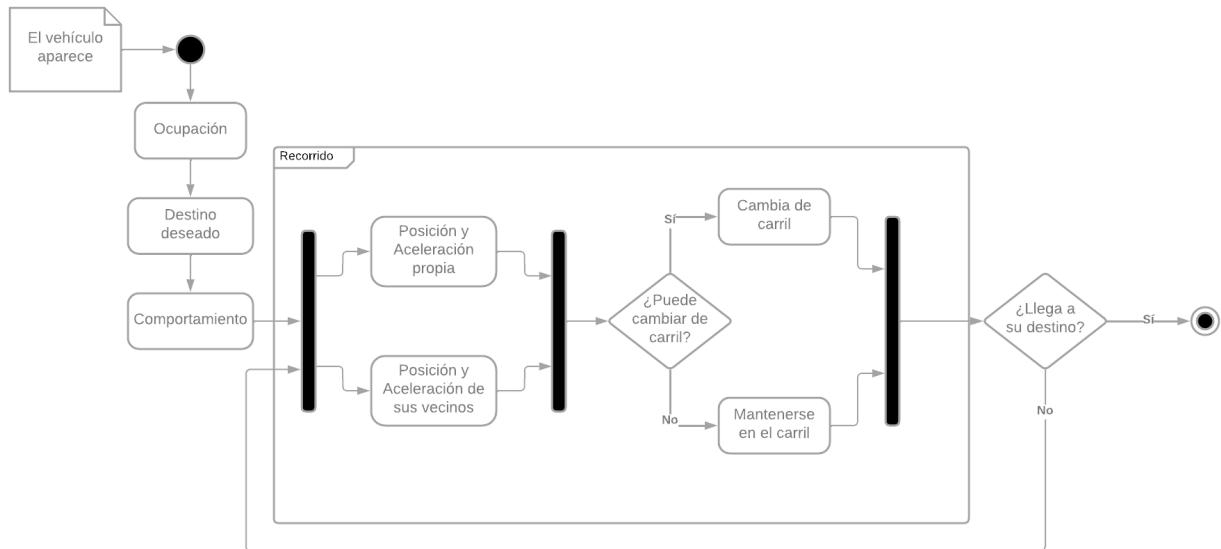


Diagrama de actividades

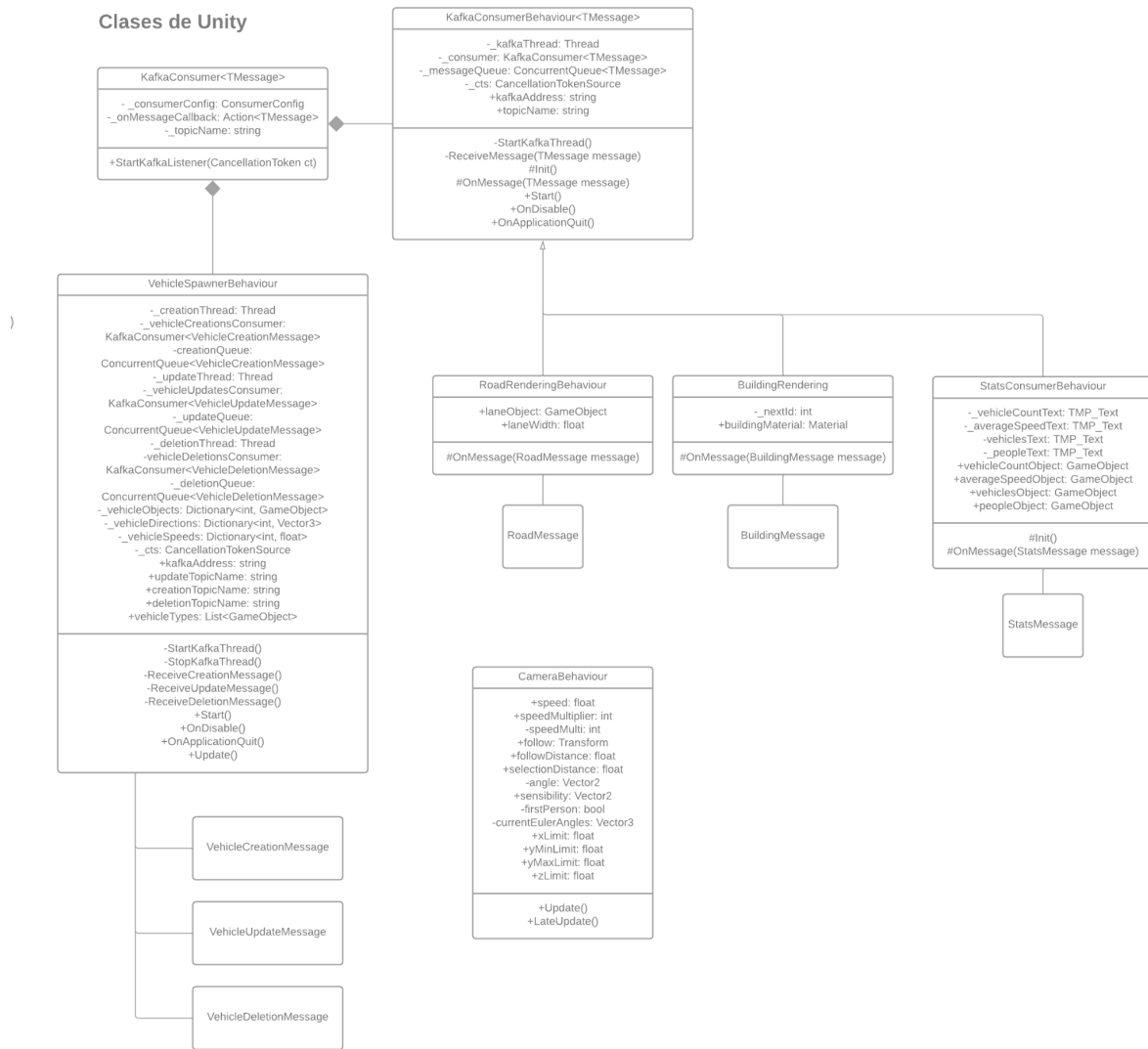


Diagrama de clases

7.1 MODELO DE LOS AGENTES

La simulación utiliza un único tipo de agente: el vehículo. Cada uno de estos agentes vehiculares inicia desde una de las posiciones de inicio preseleccionadas de la red vehicular, así como una ubicación de destino. Con esta información, los agentes calculan una ruta desde su ubicación de inicio hacia la de destino, mediante el algoritmo A* y una heurística de distancia euclidiana. Los agentes asumen que el camino con la distancia más corta es el camino más rápido. Estos también asumen que la ruta óptima no cambia a lo largo del camino.

Una vez obtenida la ruta, los agentes empiezan a avanzar a través de los caminos para llegar a su destino. Para el comportamiento en línea recta dependemos de la ecuación de aceleración dada por el *Intelligent Driver Model*, IDM. Esta ecuación modela el comportamiento de un conductor inteligente, que mide su velocidad contra la del siguiente carro, así como la distancia, una velocidad deseada, entre otros parámetros. Para nuestra simulación estos parámetros fueron distribuidos usando una distribución normal, para así simular los diferentes tipos de conductores que hay.

Para decidir si cambiar de carril o no se utiliza el criterio *Minimizing Overall Braking Induced by Lane changes*, MOBIL Este criterio busca minimizar la cantidad de frenado causada por un cambio de carril, maximizando la propia aceleración, así como la de los agentes vehiculares que nos rodean. Mediante un factor de cooperatividad, se puede ajustar que tanto un agente toma en cuenta a los demás vehículos. Al igual que con la ecuación de aceleración de IDM, este parámetro está distribuido de manera normal.

7.2 MODELO DEL ENTORNO

La red de tránsito vehicular es modelada por medio de una estructura de grafo bidireccional, con cada camino yendo en una dirección siendo un eje del grafo. Las intersecciones entre caminos son los nodos del grafo. De esta manera los vehículos pueden recorrer los caminos, y por ende la red en general.

Cada nodo del grafo cuenta con una posición, y cada eje del grafo con un número de carriles. A partir de esta información los vehículos pueden posicionarse a sí mismos en un espacio continuo tridimensional para imitar su comportamiento real.

7.3 MODELO DE LA NEGOCIACIÓN

En la simulación actual, los vehículos no son capaces de negociar entre ellos. Los vehículos únicamente pueden observar las acciones de sus vecinos y reaccionar acorde a esto.

7.4 MODELO DE LA INTERACCIÓN

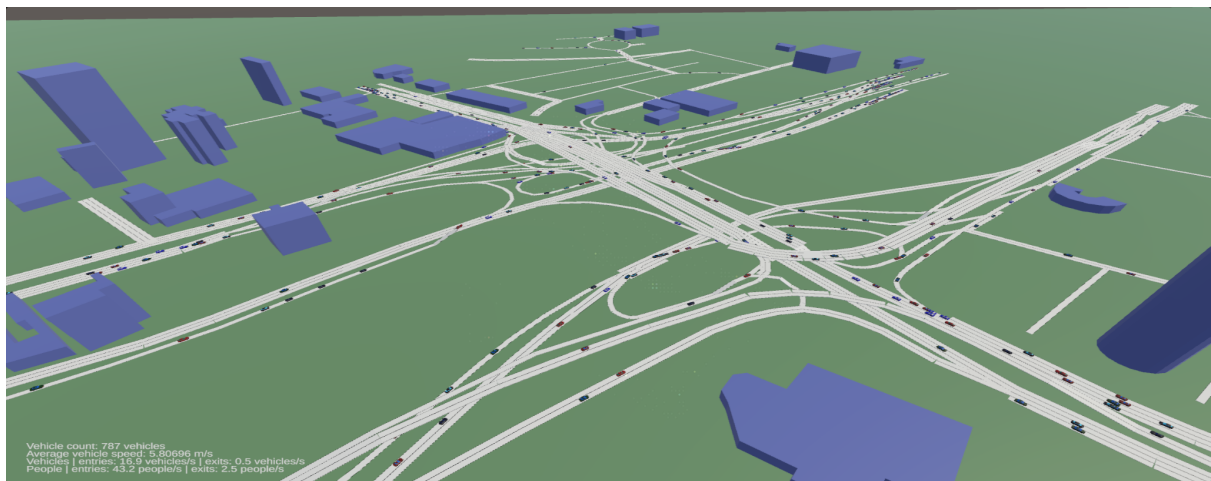
Toda la comunicación entre el módulo de multiagentes y la interfaz gráfica se logra mediante Kafka, aquí el módulo de multiagentes actualiza la información contenida en el tópico designado a la información de los vehículos. Ante esto, la interfaz gráfica de Unity se suscribe a este tópico, teniendo así acceso a la información de los vehículos y permitiéndonos representarlos de manera gráfica gracias a los diferentes prefabs dentro de Unity.

Al solo tener un agente instanciado que son los carros con diferentes modelos 3d, cada uno de ellos funciona de una manera similar.

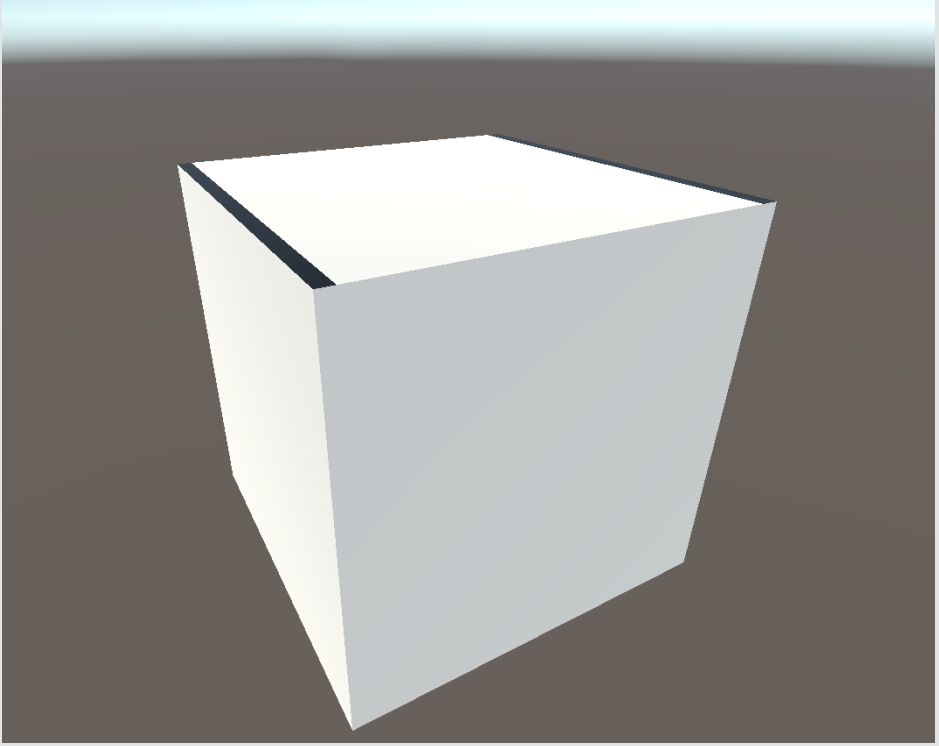
8. DESCRIPCIÓN DEL MODELADO GRÁFICO


8.1 ESCENA A MODELAR


Los caminos se generarán proceduralmente con la información obtenida de OpenStreetMap, de esta manera la simulación se puede reutilizar para cualquier sistema vehicular que esté registrado en OSM. Se le agregaran edificios a la simulación igualmente utilizando las geometrías de edificios proveídos por esta plataforma.





8.2 COMPONENTES GRÁFICOS


Nombre	Lane
Descripción	Un mesh creado para modelar los carriles mediante instancias
Imagen	
Fuente	Hiram Muñoz

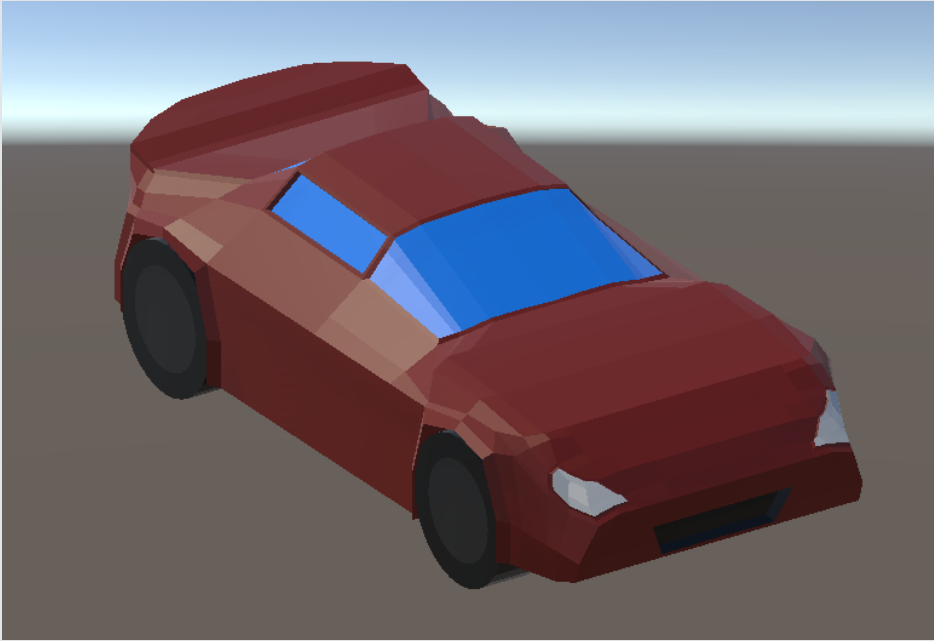
Nombre	BMW
Descripción	Modelo de un BMW en base a planos reales
Imagen	
Fuente	Angel García


Nombre	Alpina b7
Descripción	Modelo de un Alpina b7 en base a planos reales
Imagen	
Fuente	Bryan De La O

Nombre	Corolla
Descripción	Modelo de un Corolla en base a planos reales
Imagen	
Fuente	Bryan De La O

Nombre	Kia Rio Sedan
Descripción	Modelo de un Kia Rio Sedan en base a planos reales
Imagen	
Fuente	Angel García


Nombre	Nissan Versa
Descripción	Modelo de un Nissan Versa en base a planos reales
Imagen	
Fuente	Angel García


Nombre	Saturno
Descripción	Modelo de un Saturn en base a planos reales
Imagen	
Fuente	Angel García


Nombre	low_poly_audi
Descripción	Modelo de un Audi en base a planos reales
Imagen	
Fuente	Angel García


8.3 PREFABS


Nombre	Alpina b7 Variant.prefab
Descripción	Modelo tomado por el código para instanciar vehículos, le agrega un box collider al vehículo.
Imagen	
Scripts	Ninguno


Nombre	BMW Variant.prefab
Descripción	Modelo tomado por el código para instanciar vehículos, le agrega un box collider al vehículo.
Imagen	
Scripts	Ninguno

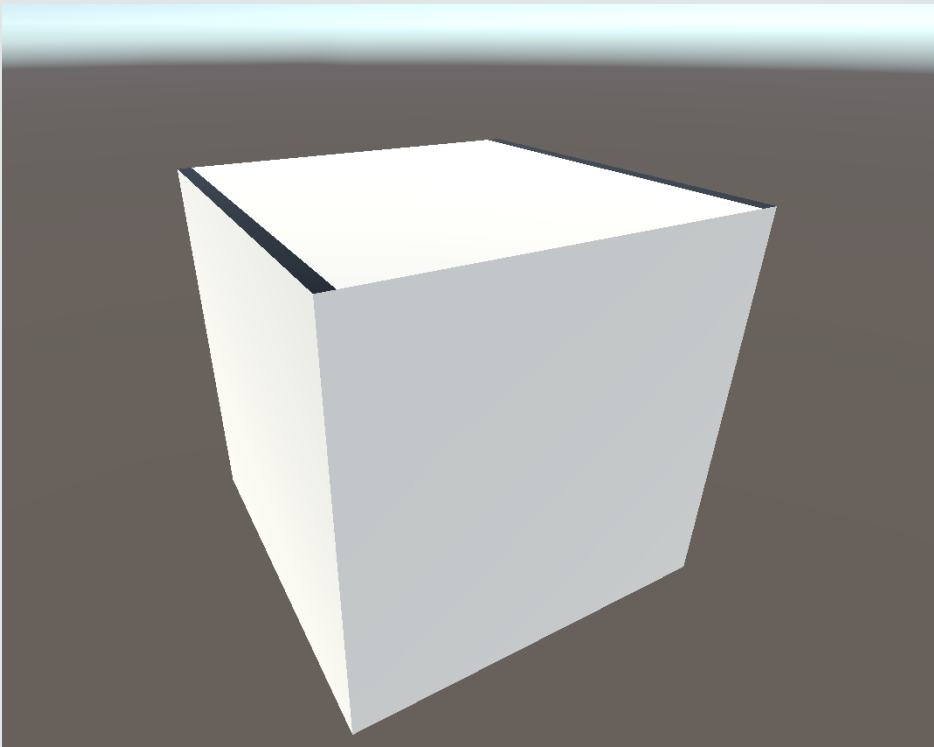
Nombre	Corolla Variant.prefab
Descripción	Modelo tomado por el código para instanciar vehículos, le agrega un box collider al vehículo.
Imagen	
Scripts	Ninguno

Nombre	Kia Rio Sedan Variant.prefab
Descripción	Modelo tomado por el código para instanciar vehículos, le agrega un box collider al vehículo.
Imagen	
Scripts	Ninguno

Nombre	Nissan Versa Variant.prefab
Descripción	Modelo tomado por el código para instanciar vehículos, le agrega un box collider al vehículo.
Imagen	
Scripts	Ninguno

Nombre	Saturno Variant.prefab
Descripción	Modelo tomado por el código para instanciar vehículos, le agrega un box collider al vehículo.
Imagen	
Scripts	Ninguno

Nombre	low_poly_audi Variant.prefab
Descripción	Modelo tomado por el código para instanciar vehículos, le agrega un box collider al vehículo.
Imagen	
Scripts	Ninguno

Nombre	Lane.prefab
Descripción	Objeto que se instancia por carril de un cierto segmento de calle.
Imagen	
Scripts	Ninguno

8.4 SCRIPTS

Nombre	BuildingRenderingBehaviour.cs
Descripción	Script que obtiene la geometría de los edificios de OpenStreetMap y genera un prisma a partir de un polígono como base.
Interacciones	Este script no interactúa con otros elementos del proyecto

Nombre	RoadRenderingBehaviour.cs
Descripción	Script que toma las calles de OpenStreetMap y las genera dentro del entorno gráfico, generando cada carril de las calles.
Interacciones	Este script no interactúa con otros elementos del proyecto

Nombre	VehicleSpawnerBehaviour.cs
Descripción	Script encargado de instanciar los vehículos, actualizar su posición y eliminarlos al salir del mapa
Interacciones	Al estar suscrito a los tópicos del servidor de Kafka, crea y destruye a los vehículos y toma sus posiciones para actualizarlas en el entorno gráfico, esta información la obtiene del sistema multiagente que actualiza la información en los tópicos

Nombre	CameraBehaviour.cs
Descripción	Script de la cámara que nos permite controlar su movimiento con el teclado y el mouse, así como fijar la cámara en un vehículo
Interacciones	A primera instancia la cámara solo se controla con WASDQE y el click derecho, pero a hacer click izquierdo en cualquiera de los vehículos del mapa la cámara se enfocará en este y lo seguirá, brindando al usuario la posibilidad de girar la cámara a su alrededor y pasar a una cámara en primera persona del vehículo

9. ENTREGABLES DE ADMINISTRACIÓN DE PROYECTO

<https://github.com/users/Stock44/projects/2/views/1>

10. REFERENCIAS

[1] M. Treiber and A. Kesting. *Traffic Flow Dynamics*. Berlin Heidelberg: Springer, 2013.

[2] GeeksforGeeks (2023) A* search algorithm, GeeksforGeeks. GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/a-search-algorithm/> (Accessed: March 15, 2023).

[3] García, G. (2023) Los 371 autos más vendidos de México en 2022: El ranking completo con todos los modelos, Autos más vendidos de México en 2022: lista completa con todos los modelos. Motorpasión México. Available at: <https://www.motorpasion.com.mx/industria/autos-vendidos-mexico-2022> (Accessed: March 15, 2023).

[4] Fabela, M.de J. and Durán, O. (2007) Resumen Boletines, Instituto Mexicano del Transporte. Available at: <https://imt.mx/resumen-boletines.html?IdArticulo=307> (Accessed: March 17, 2023).