



ITESM- Campus Puebla

**Modelación de sistemas multiagentes con gráficas
computacionales**

M1: Arranque de proyectos

Integrantes Equipo 4:

Víctor Alfonso Mancera Osorio A01733749

Uziel Humberto López Meneses A01733922

Fernando de la Torre Rodríguez A01174460

Alejandro López Hernández A01733984

Fecha: 12/11/2021

Herramientas de trabajo colaborativo

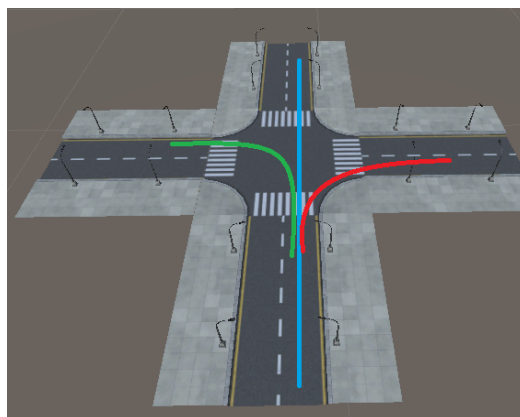
Propuesta formal

Descripción del reto

Con motivo de proponer alternativas a problemas de movilidad urbana que se suscita en grandes urbes de México, se propone implementar una simulación gráfica del tráfico que se presenta en una sección de la infraestructura que suele haber en las ciudades. A través de la implementación de un sistema modelo agente, se busca una simulación representada en Unity en conjunto de llamadas al código hecho en el framework de Python Mesa situado en el servicio de nube con el que cuenta el Socio Formador IBM. En esta ocasión, la simulación hace énfasis en emular el comportamiento de múltiples autos dentro de un cruce entre calles como el que se muestra a continuación.



En donde cada calle es de doble sentido y cada sentido cuenta únicamente con un carril. Por lo tanto, cada auto tendrá 3 opciones para transitar considerando la vista del conductor como orientación: seguir derecho (azul), doblar inmediatamente a la derecha (rojo), y doblar a la izquierda en el respectivo carril (verde).



En adición a curvas, semáforos y cruces más pequeños obtenemos la siguiente pista; adicionada con edificios que emulen puntos de llegada y de salida de los vehículos.



Las calles suponen caminos con 4 posibles puntos de partida o de llegada según en dónde se genere el automóvil y su posición actual. A pesar de contar con pocos puntos iniciales, existen múltiples opciones de camino por las cuales los automóviles puedan transitar debido a los cruces de 4 caminos situados en el centro. Se contempla que ninguno de los cruces cuente con semáforo; de tal manera que sean cruces continuos con precaución. A continuación se muestra la pista completa desde arriba, resaltando las partes importantes de las calles que son de un solo carril en ambos sentidos.



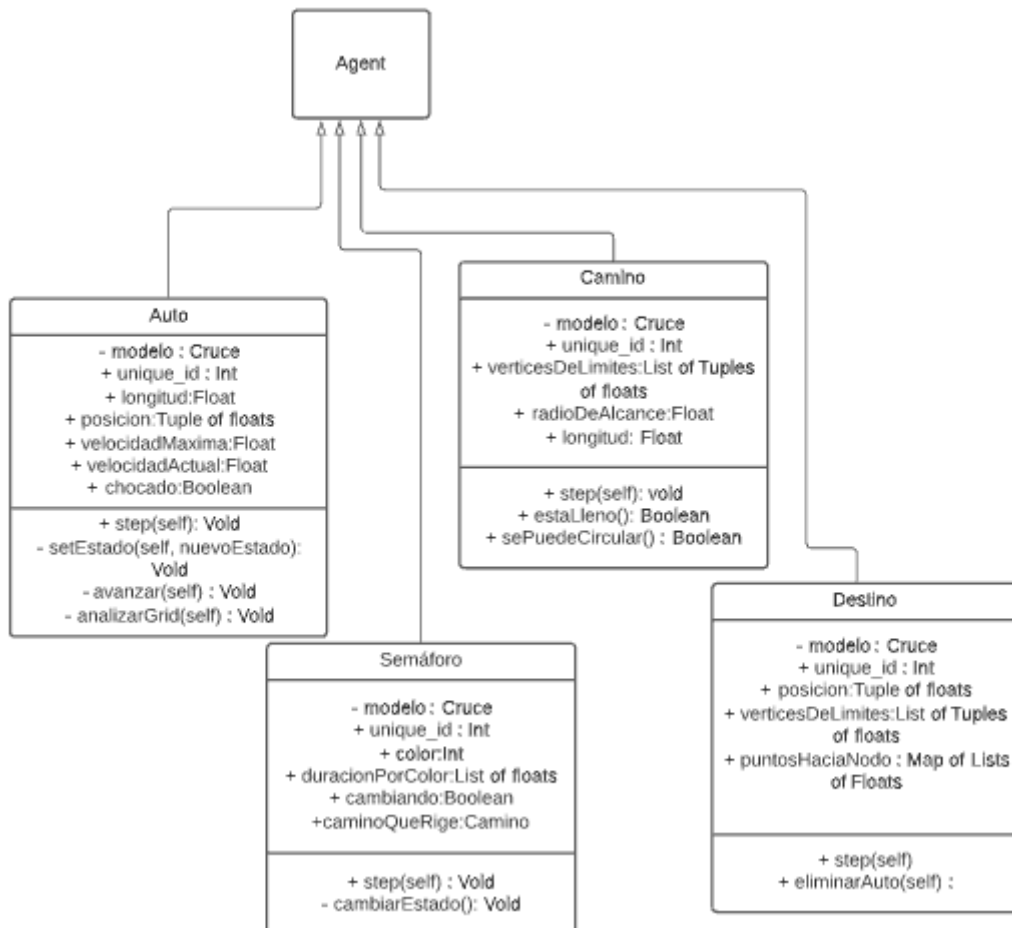
Agentes involucrados

Diagrama de clases

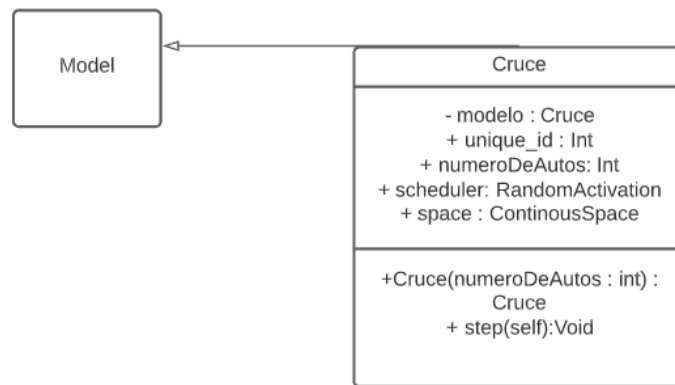
Para modelar la situación previamente descrita, proponemos la creación de un sistema de multiagentes creado con un módulo de Python llamado *Mesa*. Mesa ofrece las clases base para crear nuestros propios sistemas multiagentes como lo son la clase Agent y Model y

otras tantas que nos permiten controlar la simulación de dichos modelos como RandomActivation o BaseScheduler.

Usando estas clases, presentamos el diagrama de clases correspondiente a los agentes que formarán parte de nuestra simulación y el modelo que los encapsulará a todos.



Cada clase cuenta con variables que determinan el estado específico de cada uno de los agentes y estas variables son hechas públicas para que los otros agentes puedan tomar decisiones sobre qué hacer en cada paso de la simulación, apoyándose de sus respectivos métodos privados y públicos.



En cuanto a nuestro modelo, le proporcionamos los elementos básicos de modelado que ofrece Mesa: un *scheduler* para ejecutar la simulación en pasos discretos, una instancia de *ContinousSpace* que nos da el espacio “físico” donde colocar a los agentes utilizando coordenadas rectangulares. De igual manera, proporcionamos al constructor la posibilidad de establecer la cantidad de autos que se simularán.

En la siguiente sección, explicaremos cómo los distintos componentes de los agentes y del modelo interactúan para darle vida a nuestra simulación.

Diagrama de protocolos

Nuestro diagrama de protocolos muestra la comunicación que existe entre los distintos agentes y modelo al momento de evaluar cada una de las acciones que se ejecutan al momento de coordinar el “step” de los objetos.

Empezaremos explicando el propósito del modelo, el cual inicialmente se comunica con todos los agentes para colocarles en el Grid e inicializarlos, además de administrar los “steps” de cada uno de estos. A su vez, este posee métodos para evaluar el estado del Grid con la finalidad de brindar información al solicitante (Dependiendo del tipo de solicitud del mismo)

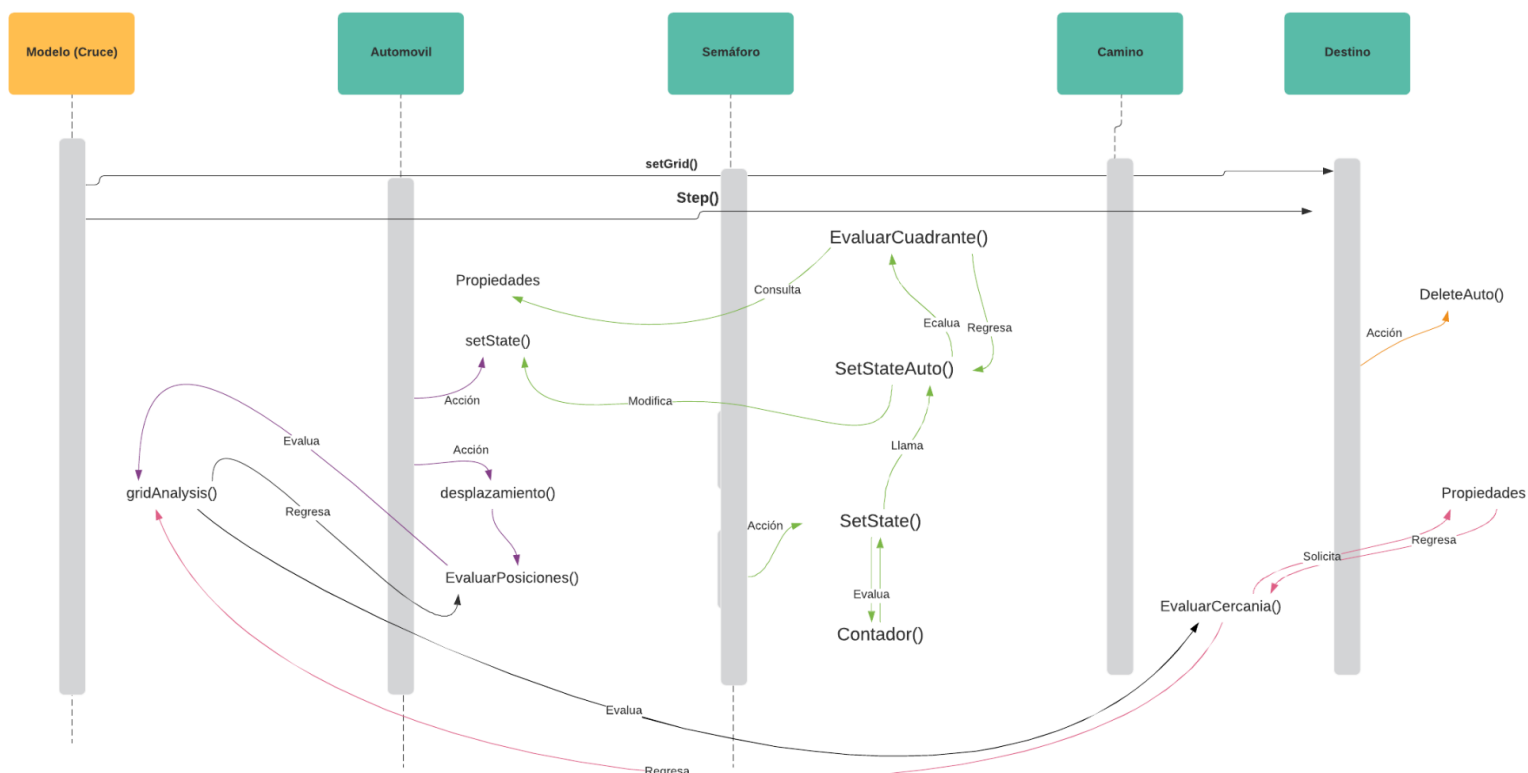
Posteriormente tenemos al automóvil, el cual primeramente tiene un método para determinar en qué estado se encuentra (SetState) permitiéndole avanzar o detenerse según sea el caso. A su vez, el método de “desplazamiento” se encarga de coordinar los movimientos del automóvil, auxiliándose del método “Evaluar posiciones” el cual a su vez, hace uso del análisis del grid para encontrar los automóviles cercanos, los caminos disponibles y su relación con el destino que le corresponde al automóvil en cuestión.

Consecuentemente tenemos al semáforo, el cual tiene el propósito de evaluar su correspondiente cuadrante con la finalidad de encontrar todos aquellos automóviles

que se ven influenciados por su propio estado. Ante ello, tenemos un método que nos permite modificar el estado de todos los vehículos en presencia de este semáforo (Determinando así si estos deben quedarse estáticos o en movimiento). Por último, tenemos el método que determina el estado del semáforo (Y que eventualmente manda a llamar la modificación del estado de los autos) que se ve influenciado por un contador que establece si este debería encontrarse en amarillo, rojo o verde.

Finalmente, tenemos los dos objetos restantes; camino y destino. El primero de ellos tiene el propósito de representar un camino viable por el cual pueden desplazarse los automóviles, además de determinar siempre su cercanía con el punto en destino a través de sus coordenadas. Finalmente, el destino es el encargado de eliminar a todos aquellos autos que hayan llegado hasta su posición.

Diagrama de protocolo de acción



Plan de trabajo y aprendizaje adquirido

- Las actividades pendientes y el tiempo en el que se realizarán.
- Realización y obtención de los modelados 3D de los agentes (automóviles).

- Realización u obtención de las texturas para el ambiente.
 - Modelado de la zona de cruce (calle), para los automóviles, con sus respectivas texturas.
- **Para las actividades planeadas para la primera revisión, los responsables de llevarlas a cabo, la fecha en las que las realizarán y el intervalo de esfuerzo estimado.**
- **Descripción del reto a desarrollar.**
 - Breve texto sobre la explicación del texto, de que va a tratar, los alcances que esperamos que estos tengan; así como el objetivo final del mismo.
 - **Encargado:** Alejandro Lopez Hernandez.
 - **Creación de herramientas para el trabajo colaborativo.**
 - Subir el código inicial de Unity a un repositorio GitHub; donde se subirá el trabajo colaborativo, del modelo y agentes; así como la lógica de los mismos implementados en Mesa.
 - **Encargado:** Fernando de la Torre Rodríguez.
 - **Diagrama de clase presentando los distintos tipos de agentes involucrados.**
 - Esquema de las clases
 - **Encargado:** Uziel Humberto Lopez Meneses.
 - **Diagrama de protocolos de interacción.**
 - Esquema que representa la secuencia de interacciones de los involucrados; agentes; con el modelo; calle; los tiempos que estos tendrán para realizar las acciones, así como las reacciones que estos presentaran al interactuar entre ellos.
 - **Encargado:** Víctor Alfonso Mancera Osorio.