

Modelación de Sistemas Multiagentes con Gráficas Computacionales

Gpo. 4

P4 Revisión de avance 2

Equipo:

Hector Guapo Guerrero	A01197463
Iván David Manzano	A01029111
Iván E. Ramírez Martínez	A01197409
Carlos Sevilla Silva	A00826925

Profesores:

Edgar Covantes Osuna

Jorge Mario Cruz Duarte

© 2020 Derechos reservados: Ninguna parte de esta obra puede ser reproducida o transmitida, mediante ningún sistema o método, electrónico o mecánico, sin conocimiento por escrito de los autores.

Monterrey, Nuevo León, 13 de agosto de 2021.

Conformación de Equipo

Iván Ramírez

Fortalezas

Las fortalezas que puedo aportar al equipo creo que es mi buena organización de trabajo. Me considero una persona responsable por lo que considero que mi trabajo esté realizado en tiempo y forma. Por la parte de conocimientos computacionales, me considero hábil para Unity así como para los lenguajes en C, así también mi experiencia con Programación Orientada a Objetos y en Estructuras de Datos.

Áreas de Oportunidad

Considero que mis áreas de oportunidad son mi tolerancia y paciencia, puedo tender a desesperarme si lo que me propongo no sale aunque lo intente. Creo que también mi falta de comunicación es algo que debo mejorar para este proyecto para tener éxito. Por último, claramente están mi poca experiencia en Python así como con los frameworks y librerías nuevas que estaremos utilizando para resolver el reto.

Expectativas

Principalmente es que nuestra propuesta de solución sea un éxito, ante todo quisiera salir satisfecho con el producto final que realicemos. Más aparte quisiera pulir mis habilidades con Python y las nuevas librerías y herramientas que se nos presenten en el curso. Como propósito para este proyecto también espero desarrollar mi paciencia tolerancia y resiliencia puesto que estoy consciente que para realizar este proyecto hará falta de bastante investigación y lectura y es una meta personal el mejorar en este aspecto en particular.

Iván Manzano

Fortalezas

Por parte de mis fortalezas considero que tengo un buen trabajo en equipo, así como trabajo personal. Ya por parte del proyecto soy bueno en Unity (level design) y C++, también soy bueno redactando y arreglando documentos. Si es necesario puedo actuar como líder, pero me conformo más con trabajar en lo que me asignen.

Áreas de Oportunidad

Tardo mucho en entender un problema. Además, tengo fallas en la comunicación con otras personas al inicio. Por último, estoy muy oxidado en Python (lo cual se necesitará para el proyecto)

Expectativas

Mi mayor expectativa es poder aprender mucho de este curso y de los profesores que hay en Campus MTY. Llegar a analizar críticamente un problema de la vida real y poder plasmarlo en código es algo que quisiera aprender. Por último, espero que este proyecto sea un éxito junto a mis compañeros de grupo.

Héctor Guapo

Fortalezas

Como fortaleza me considero una persona responsable que sabe que muchas veces va a ser necesario hacer un esfuerzo extra para poder salir adelante en los proyectos. Siempre estoy dispuesto a aprender más cosas y ser corregido si cometo errores.

Áreas de Oportunidad

Muchas veces aun que quiera aprender cosas nuevas me puede costar un poco de tiempo para poder entenderlo lo necesario para poder aplicarlo en el proyecto.

Expectativas

Espero que entreguemos un proyecto funcional que cumpla con los requisitos y aprender acerca de estas tecnologías. Ya que es bastante interesante y creo que pueden ser útiles para mi futuro.

Carlos Sevilla Silva

Fortalezas

Soy bueno adaptándome a nuevas formas de trabajo, con esto me refiero a que, con una buena explicación, puedo entender nuevo tipo de código y empezar a trabajar de forma efectiva con mis compañeros de equipo. Me considero hábil con C ++ y derivados, también entrego los trabajos en tiempo y forma.

Áreas de Oportunidad

Mientras soy bueno entendiendo el proceso de algo, puede que tarde un poco al momento de realizarlo. Eso no significa que no lo entregue, solo significa que tardará un poco más de lo normal. Esto se debe a que me distraigo muy fácil. Soy malo manteniendo liderazgo por lo que suelo dejarle eso a otras personas.

Expectativas

Espero que hagamos un trabajo efectivo en este proyecto. Confío en las habilidades de todos y estoy seguro de que lograremos una correcta implementación de las herramientas que aprenderemos en el semestre para llegar a nuestra meta. Espero poder usar esta experiencia para seguir adelante y trabajar en mis áreas de oportunidad.

Logros Esperados y Compromisos para Lograrlo

Como equipo comprendemos que nuestro principal objetivo para este reto y curso es desarrollar un proyecto exitoso que sea funcional y que sea de valor añadido para nuestra enriquecimiento como profesionales. Esperamos de este proyecto desarrollar habilidades nuevas así como aprender a utilizar herramientas, frameworks nuevos que podamos incluir en nuestro abanico de conocimientos computacionales. Como compromisos para lograr nuestros objetivos nos comprometemos a ser perseverantes y resilientes a los obstáculos que puedan presentarse durante el desarrollo del reto, así también nuestra dedicación y disponibilidad para aprender nuevas cosas ya sea investigando o aprendiendo directamente de los asesores para este proyecto.

Creación de herramientas de trabajo colaborativo

Para manejar los avances de nuestro proyecto se decidió usar un repositorio en GitHub. En ese repositorio publicaremos los avances de código y tareas del proyecto individuales para tener evidencia del trabajo.

Link: <https://github.com/IvanRmz09/MultiagentesCIA.git>

Glosario

A continuación se detalla terminología importante para el resto de documento y comprender el significado cuando se nombre algún concepto. Qué es un fw, libere, agente,

Término	Definición
Modelo	Modelo de interacción entre agentes
Agente	Sistema informático que es capaz de realizar una acción independiente en nombre de su usuario o propietario
Framework	Entorno de trabajo donde se tiene estandarizados ciertos conceptos, prácticas y criterios.
Librería	Conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.co
Jupyter Notebook	Formato de documento basado en JSON
Google Collaborative	PLataforma en línea para editar el Jupyter Notebook.
Función	Conjunto de instrucciones que resuelven un proceso específico.
PEAS	Estándar para describir un agente
Mesa	Framework de modelación de agentes
Singlegrid	Implementación del framework mesa. Permite solo a un agente ocupar una casilla.
Multigrid	Implementación del framework mesa. Permite a múltiples agentes ocupar una casilla.
Escuchar	Esperar una respuesta de otro agente
Informar	Enviar un estatus de algún parámetro solicitado por otro agente
KQML	Protocolo de Comunicación entre agentes

Descripción de la Situación

La situación que se presenta es la de diseñar un modelo donde se simula una intersección de avenidas controlado, es decir que tiene semáforos, de 4 avenidas. Una intersección se define por un cruce de 2 o más avenidas las cuales se intersectan entre sí en un punto medio de todas las calles que la componen. Cada una de las calles que la componen tiene como mínimo 2 carriles, 1 de entrada y otra de salida de la intersección. De acuerdo a las leyes de tránsito y la ubicación donde pongan dicho cruce el programa de luces para los semáforos puede variar. Podría permitir que 2 semáforos estén en verde para permitir transitar 2 líneas, aunque estas deben ser estrictamente en sentidos opuestos o permitir que solo 1 semáforo esté en verde y permitirle al conductor tomar cualquier salida que pueda. Para este caso tomaremos el segundo ejemplo para realizar el modelo. Además, se establece que la intersección se compondrá de 4 avenidas, cada una compuesta por 2 carriles: 1 de entrada a la intersección y el otro de salida. La intersección también contará con 4 semáforos, donde como ya se mencionó previamente, solo 1 podrá ponerse en verde mientras que los otros se encuentran en rojo. Los carros entran por el carril derecho de la avenida y avanzan por su carril a velocidad constante respetando la velocidad máxima permitida. El carro deberá frenar en la esquina de la intersección si el semáforo se encuentra en rojo. Además, debe tomar en consideración los carros del entorno, debe respetar la distancia mínima recomendada que debe haber automóviles. Los carros desaceleraran su velocidad gradualmente hasta llegar a la esquina si el semáforo se encuentra en amarillo, pero también lo realizan si tiene un carro en frente que está parado o cambiando de dirección. En resumen, Los conductores son responsables de reaccionar a su entorno que son otros conductores, los semáforos y peatones para avanzar y no chocar con ningún vehículo, respetar las señales del semáforo y no atropellar peatones.

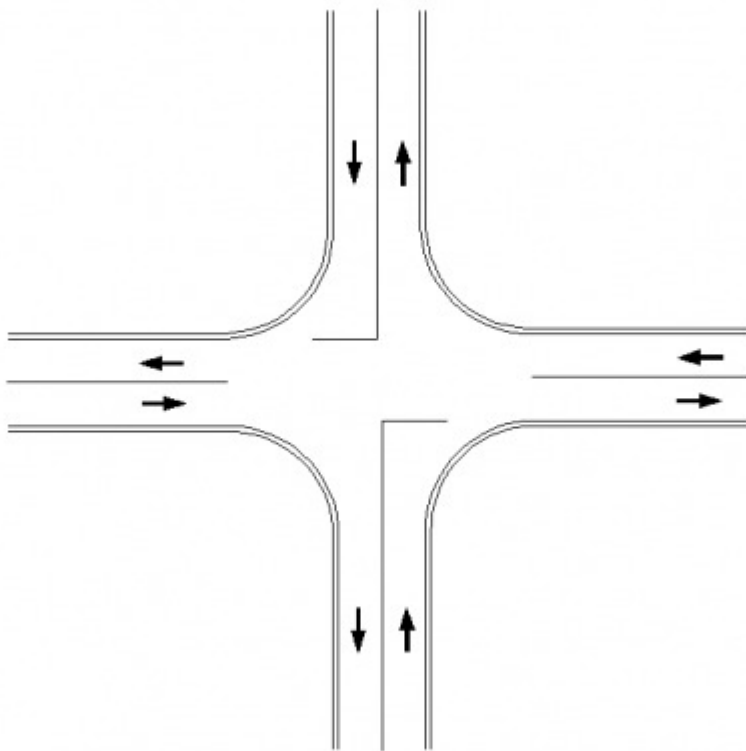


Fig 1. representación de la intersección con 4 avenidas cruzadas.

Esta situación suele ser bastante caótica por múltiples motivos dentro de los cuales se encuentran: tráfico, y choques automovilísticos que conducen a poner en riesgo la vida de las personas. Esta situación es agravante pues se calcula que aproximadamente el 40% de los accidentes viales suceden en cruces o intersecciones. Esto implica congestiones de vehículos que vuelven estos puntos de encuentro entre avenidas lentas e interfieren con las tareas diarias de las personas.

La propuesta formal del reto

Nuestra propuesta se enfoca en solucionar el problema de movilidad en un cruce de avenidas. Es bien sabido que muchos accidentes viales, así como congestiones automovilísticas son causadas en este tipo de cruces puesto la gran cantidad de vehículos que transitan por hora, cultura vial, y demás factores.

Para desarrollar este proyecto serán requeridos las siguientes herramientas. Para comenzar, se desarrollará en el lenguaje de programación Python, en su versión 3 o más reciente. Se necesitará específicamente esas versiones debido a que se utilizará el framework mesa, el cual nos permitirá desarrollar el modelo con multiagentes. Además, se requerirán las siguientes librerías, numpy, panda y matplotlib. Las primeras 3 serán para el manejo de matrices, arreglos, manipulación matemática, etc. La última librería, matplotlib, será necesaria para tener una visualización gráfica del modelo. Todo esto se utilizará dentro del entorno de trabajo de una Jupyter Notebook controlada desde la herramienta Google Collaborative. Ni el framework ni librerías serán necesarias de descargar puesto a que todo se alojará en la nube y el propio servicio se encarga de instalar las cosas necesarias para este funcionar. Cabe destacar que sí es necesario declarar la importación de librerías y framework para que estos estén implementados en el proyecto.

A continuación se describe el concepto en qué consiste cada agente de acuerdo a la descripción PEAS.

Agente	P (Performance)	E(Entorno)	A (Actuadores)	S(Sensores)
Coche	Velocidad promedio en la calle, tiempo para cruzar la intersección.	Avenida, otros vehículos, peatones, semáforo	Dirección, acelerador, freno.	Focos del Semáforo, Retrovisor, luces traseras de los vehículos
Semáforo	Tiempo de luz verde por semáforo, cantidad de carros parados en un rojo,	Avenida, vehículos, otros semáforos.	Focos	Cámara, señal de los demás semáforos

El coche se define como un agente racional reactivo simple. Este solo reacciona a las condiciones establecidas. Sus condiciones se definen como: Si hay un coche en frente, frena. Si ve el foco del semáforo en rojo de la calle que transita, debe frenar. Si ve un foco naranja

en el semáforo de la calle que transita, va desacelerando. Si ve un foco verde en la calle que transita, acelera hasta la máxima velocidad permitida y posible para avanzar en la dirección que tomó. Otra condición es tomar una dirección de la intersección. Como se observa realmente el carro solo debe reaccionar al entorno con las reglas que se le establecen desde un inicio.

Por otro lado, el semáforo es un agente basado en objetivos. El objetivo del semáforo es dejar la avenida vacía, sin carros. Para esto existen varias condiciones. Mientras el semáforo no detecte carros transitando en su avenida, este tendrá el foco en amarillo. Si el semáforo capta a través de la cámara un vehículo en la calle que le corresponde, este envía a la red de semáforo su solicitud como avenida transitada. Esto la mete dentro del programa de luces para los semáforos que tengan este estatus de transitada. Cuando el semáforo ya no detecta más carros en su avenida esta enviará la solicitud para salir del conjunto de semáforos en estado de transitada y retornará al foco color amarillo. El semáforo medirá cuantos carros cruzan desde su avenida, esto le servirá para guardar en su memoria el número de carros que transitan por hora esta avenida. Esto servirá para ajustar los tiempos que deben mostrar cada semáforo en verde, es decir, después de la actualización de información en la red de semáforos, el programa de luces se reajustará para distribuir el tiempo que tienen los semáforos para ponerse en verde de acuerdo a la prioridad en la que los semáforos están, con base a la densidad de carros que cruzan la avenida por hora.

Entre algunas de las funciones que se pueden identificar para que los agentes se encuentran:

- Coche: Avanzar(): esta es una simple acción de movimiento del agente que le permite moverse a través del modelo en la dirección que permite la distribución vial (no se salga de la calle o de alguna vuelta ilegal). Esta acción le permite alcanzar la velocidad máxima permitida por la ley mientras que su entorno se lo permita. Revisará el estatus de su entorno, y el factor de aceleración cambiará dependiendo del estatus del entorno.
- Coche: Desacelerar(): Acción de movimiento que reduce la distancia recorrida en el tiempo gradualmente hasta llegar a 0 m/s de ser necesario. El factor de desaceleración variará de acuerdo al estatus del entorno.
- Coche: RevisarEntorno(): Como su nombre lo dice, el agente o che revisará su entorno para ver si hay más coches alrededor del mismo, también obtendrá la distancia que lleva en relación al coche de enfrente. Además, revisará el estatus del semáforo en la calle que se encuentra. Esto al final almacena una instrucción que de acuerdo a condiciones como cercanía del vehículo y estado del semáforo dará un factor positivo o negativo que determina si debe acelerar (si el factor es positivo) o desacelerar (si el factor es negativo).
- Semáforo: DetectarCarro(): El semáforo detecta cualquier carro y envía la solicitud para entrar a la cola de semáforos transitados.
- Semáforo: CambiarColor(): Se sincroniza con el programa de luces de la red de semaforos transitados y revisa si es su turno y el tiempo. Si es su turno y tiene un tiempo mayor a 30 segundos, cambia el estado del foco a verde, y se actualiza. Cuando detecte que es su turno y le quedan menos de 30 segundos, se tornará amarillo. Finalmente cuando detecte que no es su turno se tornará rojo.
- Semáforo: StandBy(): Cuando la cámara no detecte ningún vehículo en su calle, enviará la solicitud para salir de la cola de semáforos transitados. Cuando reciba la señal de recibido y aceptado, el estado del color del foco se cambiará a amarillo. y se dejará la cámara activada para detectar cuando otro coche cruce.

Se identifican varios conceptos importantes dentro de los métodos. Para el caso del agente coche, es importante notar que en sus funciones lo primordial es el escuchar las señales del entorno y responder con base a ellas por medio de acciones. Es decir, estas son llamadas de pedido, necesitan saber la respuesta de los demás agentes que lo involucran para poder continuar y tener una respuesta, que a su vez sirve de información para los demás agentes. En el caso del semáforo es más complejo, involucra el informar a los otros agentes, tanto semáforos como coches, así como solicitar una respuesta. Las funciones de semáforo también esperan respuesta de agentes del entorno para cambiar el estado del semáforo. Cabe destacar que las funciones se hicieron tomando en cuenta el protocolo de comunicación KQML, de ahí se comprende que las funciones *pidan* información, *realizan* acciones, *informan o dicen* sus estados a los agentes que solicitan esa información o a todos por igual.

Explicación del Modelo

El modelo establece que se compone de un grid el cual no es cíclico y se compondrá de casillas. En estas se encontrarán los agentes. Es importante definir que para crear el modelo debe de establecer en cuáles celdas se puede permitir transitar a los agentes de tipo coche, así como en cuales se ubicarán los agentes de semáforo, donde además, pueden compartir espacio con los coches, es decir será Multigrid en las celdas de semáforos en las celdas que representan la calle. El modelo establece una relación colaborativa entre los agentes de tipo coche y otra de tipo semáforo. El objetivo de todos es eficientizar el tiempo dentro del cruce de los vehículos. Esto se obtiene permitiéndole el verde a semáforos con mayor prioridad por densidad de tráfico que la cruza, así como que semáforo recibió primero coches.

Para esto es importante notar que los semáforos se encontrarán almacenados en una estructura de datos tipo Queue. Esta pondrá en un listado en el orden de entrada a la red de semáforos transitados a los semáforos de la intersección. Cuando un semáforo se desocupa desaparece de esta lista. La razón para utilizar esta estructura de datos, es para determinar el orden de luces verdes que se le permitirá dentro del programa de luces a los semáforos, como se explicó anteriormente. Es importante para que en el modelo se implemente un sistema de aprendizaje en el cual se aplique posteriormente al programa de luces, para que este sea más eficiente gradualmente. También es importante limitar las casillas de los vehículos en movimiento y representar distancias con estos, de acuerdo a la planeación, cada casilla tiene el valor estándar de 3 metros, a excepción de las que ocupan los semáforos. Entre vehículos se debe dejar por lo menos una celda de distancia y en el semáforo debe parar una celda antes.

Diagramas

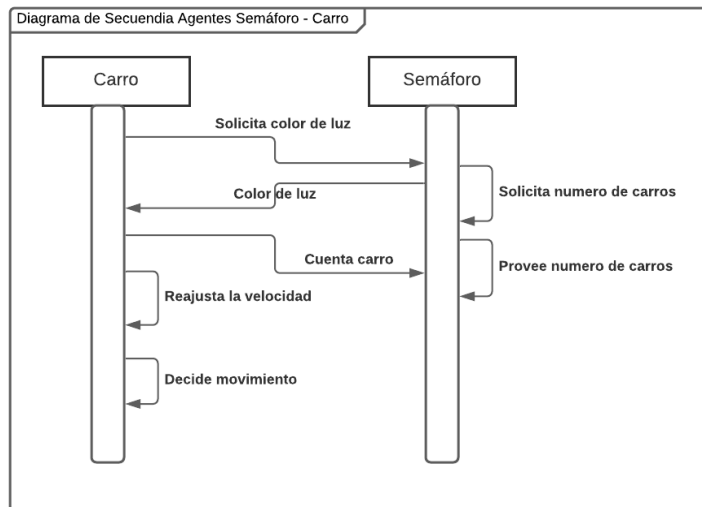


Fig 2. Diagrama de clases entre los agentes

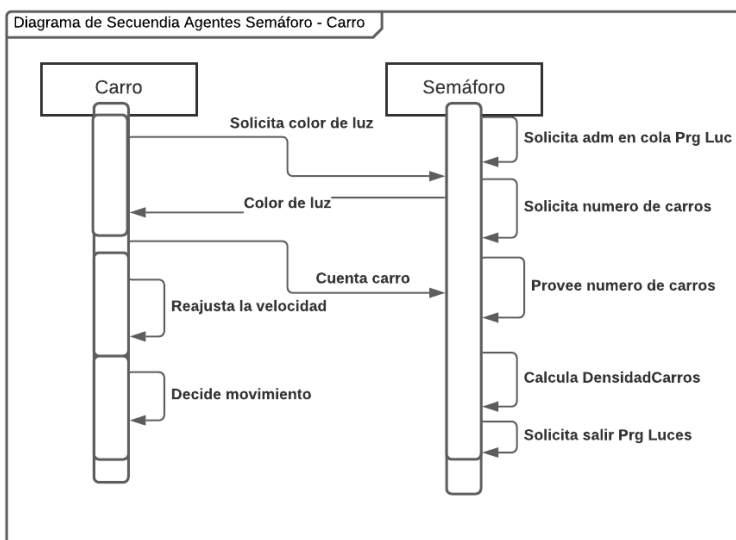


Fig 3. Diagrama de Secuencia de los protocolos entre agentes.

Plan de trabajo

Actividades realizadas:

- Introducción de documentación / Arranque del proyecto.

Actividades pendientes:

- Diagrama de requerimientos Encargados: Hector Guapo | Esfuerzo: 12 horas/persona
- Desarrollo de mapa en Unity Encargados: Iván Manzano| Esfuerzo: 6 horas/persona
- Desarrollo assets de agentes en Unity Encargados: Iván Manzano, Iván Ramírez| Esfuerzo: 24 horas/persona
 - Prefab automóviles
 - Semáforos
 - Calles
 - Espacio
- Desarrollo de acciones de agentes Encargados: Todos | Esfuerzo: 36 horas/persona
 - Cambio de Color: Semáforo
 - Establecer programa de luz: Semáforo
 - Movimiento: Automóvil
 - Observar el semáforo: Automóvil
- Desarrollo del modelo Encargados: Todos | Esfuerzo: 72 horas/persona
 - Desarrollar las clases de agentes (Semaforo y Automóvil)
 - Desarrollar el modelo con las reglas propuestas
 - Pendiente definir más actividades
- Integración Final de las partes Encargados: Todos | Esfuerzo: 12 horas/persona
 - Integrar modelo funcional a Unity

Planes de Mejora

Para este proyecto quedan aún muchas áreas de oportunidad. Es necesario para el modelo definir qué modelo de aprendizaje adoptará y también los criterios. En las mejoras a corto plazo debe aprovechar esta capacidad de memoria y calcular la densidad de carros para así volver más preciso el programa de luces para la intersección. Finalmente, Implementar en el modelo un plan de contingencia por si ocurre un accidente, este modelo se construyó omitiendo peatones y accidentes potenciales que ocurran, es lo más importante para mejorar este proyecto y que llegue a ser una solución realista a lo que el entorno real representa.