

Tecnológico de Monterrey

CII.IA Entrega Final

Profesores

Raúl Valente Ramírez Velarde

Luis Alberto Muñoz Ubando

Unidad de formación

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 1)

Eq. 1

A00827463 Sebastián Rodríguez

A00517149 Gabriel Martínez

A00829356 Juan Fernando Fernández

Documentación: “Arranque de Proyecto”

Expectativas del curso

Dado a los temas que hemos estado trabajando lo que va del curso, como equipo contamos con expectativas de poder manejar de una manera bastante próxima a lo profesional la herramienta de Unity principalmente, siendo que ésta es la herramienta principal con la que estaremos trabajando. Además, deseamos poder aprender un poco sobre cuestiones de IA, siendo que se debe de saber lo más básico para lograr que agentes accionen por sí mismos dentro de un ambiente. Algunos conocimientos extras que nos gustaría obtener son: aprender sobre el uso de las matemáticas para lograr el movimiento fluido de objetos, aprender a renderizar aunque sea a un bajo nivel y reforzar nuestros conocimientos sobre la programación en C#.

Estamos completamente conscientes de que para lograr dichos cometidos también es necesario que nosotros pongamos de nuestra parte para aprender. Es por esto que, como equipo, nos comprometemos a mostrarnos participativos en todo momento, además de tratar de incorporar todas las implementaciones posibles a nuestro proyecto, y no quedarnos con un producto que únicamente cumpla con lo mínimo.

Propuesta Formal

Descripción del reto a desarrollar.

Dentro de este reto de movilidad urbana, lo que tenemos contemplado implementar es sincronizar los semáforos en las intersecciones para permitir así un flujo más constante de los automóviles. Del mismo modo implementar un formato sensorial en los semáforos que permita identificar si en alguna intersección no hay carros, que permita hacer el cambio en los semáforos para dar prioridad en el momento a los que están esperando en la intersección opuesta. Modificando los semáforos y utilizándolos como agente, se puede facilitar la movilización de los vehículos haciendo los tiempos de espera en el tráfico más bajo, y el camino más rápido.

Agentes involucrados.

Por definición, los sistemas multiagentes es un sistema compuesto por múltiples agentes inteligentes que interactúan entre ellos. En este caso, para desarrollar el proyecto actual, tendremos 2 clases de agentes en juego: Estos son los automóviles y los semáforos. El medio donde cada uno de nuestros agentes se desarrollarán, será una pequeña ciudad que simulará el congestionamiento vial de una gran metrópolis. Como ya lo hemos descrito, el primero de nuestros objetos será el automóvil, el cual circulará por toda la ciudad y su movimiento dependerá de 3 factores, el estado del semáforo posicionado en su vía actual, el estado y ubicación del resto de vehículos transitando por la misma calle, y por último, el algoritmo que indicará hacia dónde se dirigirá el automóvil.

Nuestro segundo agente, será un semáforo. El semáforo consta de 3 estados, verde, amarillo y rojo, cada uno de los colores controlará el movimiento de los automóviles y al mismo tiempo se coordinará con el resto de agentes del mismo tipo para generar una circulación efectiva.

Cada uno de los automóviles tendrá propiedades físicas, de manera que tendrán que resguardarse unos de otros procurando mantener una distancia que les permita maniobrar. Además, tendrán que respetar la posición de cada uno de ellos para considerar su movimiento.

Plan de trabajo

Pendiente	Responsable	Fecha estimada	Esfuerzo	Estado
-----------	-------------	----------------	----------	--------

Diagrama Inicial de escenario	Todos	12/11	Bajo	Cumplido
Investigación de waypoints	Fernando	13/11	Bajo	Cumplido
Investigación sobre el desarrollo de los automóviles	Fernando / Gabriel	14/11	Bajo	Cumplido
Implementación de waypoints en Unity	Fernando	15/11	Medio	Cumplido
Desarrollo y movimiento de los carros	Fernando / Gabriel	17/11	Alto	Cumplido
Juntar el movimiento de los autos con los waypoints	Fernando / Gabriel	20/11	Alto	Cumplido
Implementación de calles/avenidas	Sebastian	21/11	Bajo	Cumplido
Desarrollo de atributos Visuales	Sebastian	23/11	Medio	Cumplido
Incorporación de las calles con los waypoints	Todos	25/11	Medio	Cumplido
Desarrollo de los semáforos	Fernando / Gabriel	28/11	Alto	Cumplido

Documentación: “Entrega Final”

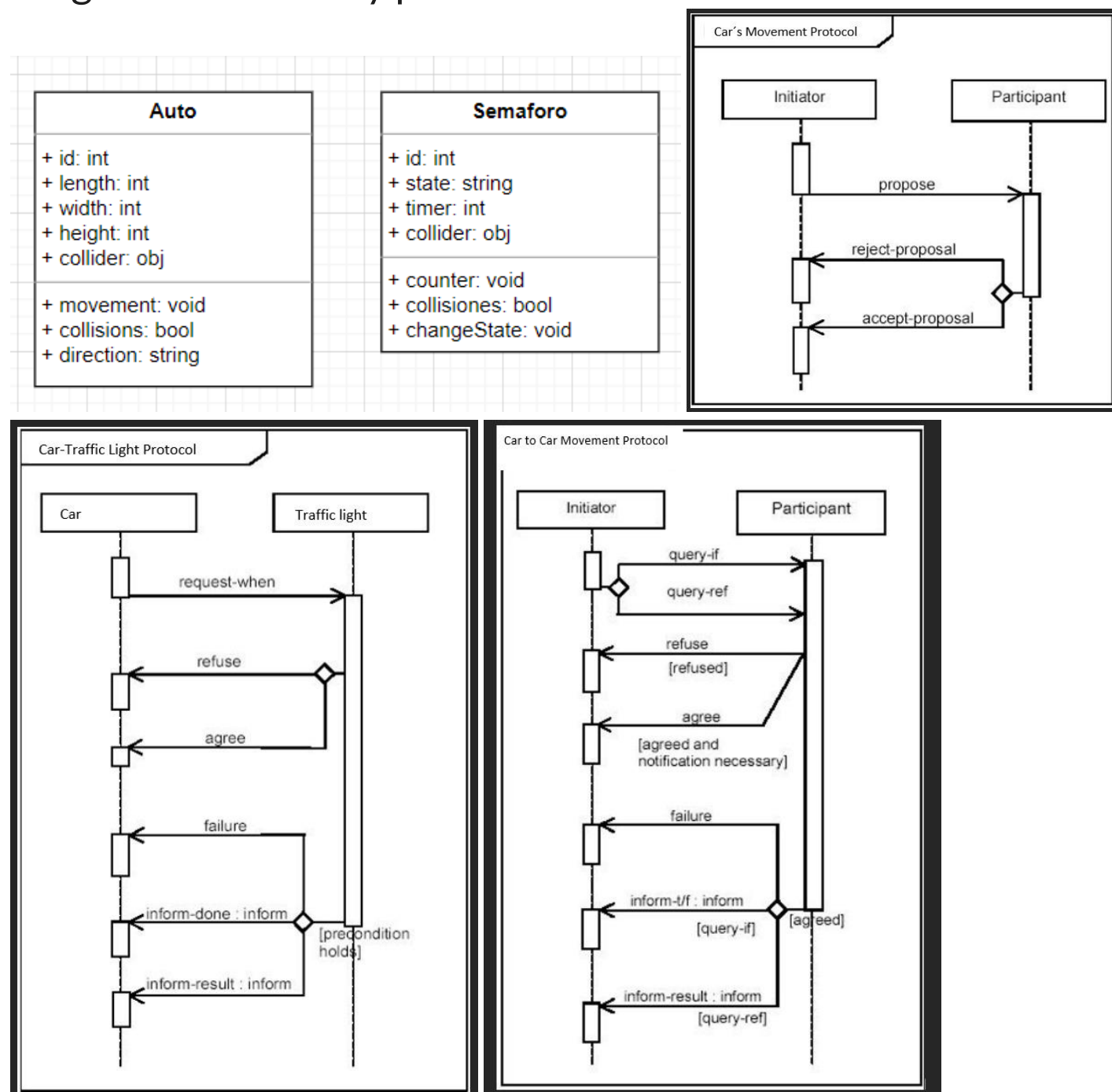
Proyecto Completo

https://github.com/A00827463/movilidad_urbana

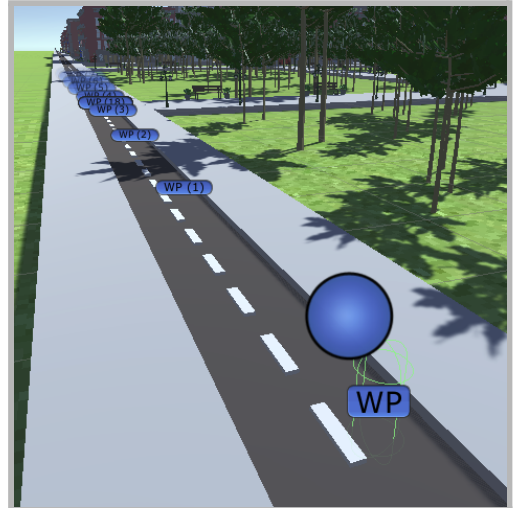
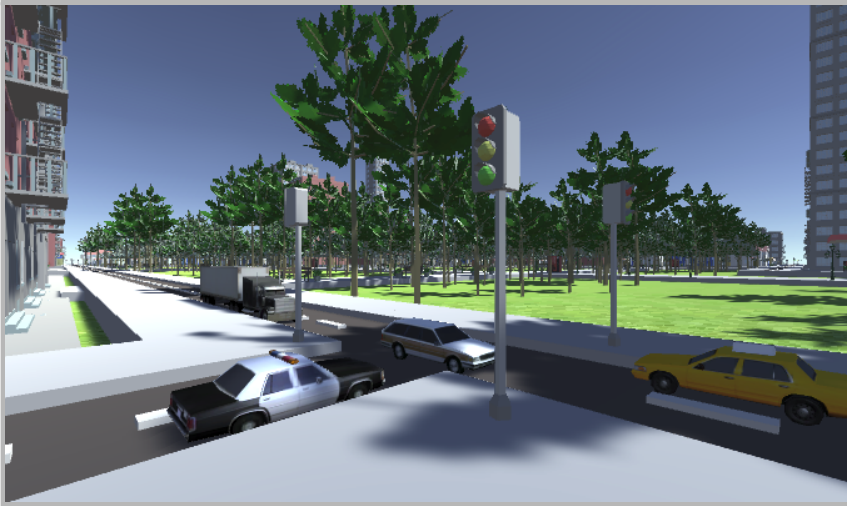
Presentación de Proyecto

<https://docs.google.com/presentation/d/1RKTkn25-a-L0hCaxALYcuJ2wUk0JcmV9PHpegKnEOMM/edit?usp=sharing>

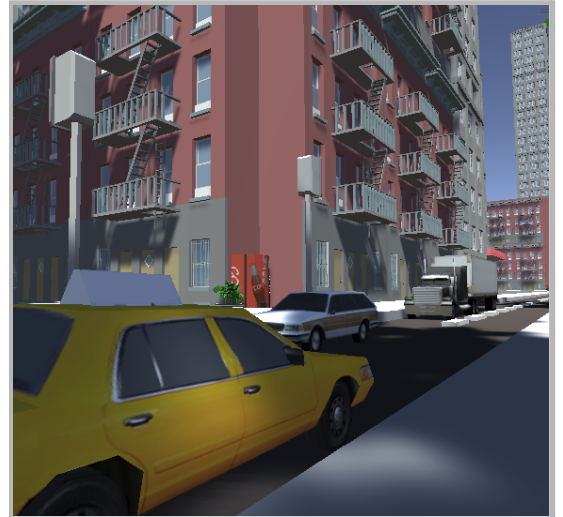
Diagramas de clase y protocolos de interacción finales.



Implementación completa de los agentes.



Implementación completa de la interfaz gráfica de la simulación.



Proceso de instalación, configuración y ejecución de la simulación. Video explicativo [aquí](#).

Unity	<p>Para propósitos de este proyecto, descargamos el plan personal y gratuito de Unity.</p> <ol style="list-style-type: none">1. Ingresar a la tienda de descarga de planes de Unity.2. Dar click en el botón de comencemos bajo el apartado Personal.3. Estando aquí, bajamos al apartado de <i>Empieza a crear con Unity en tres pasos</i>.4. Estos son los tres pasos que hay que seguir para empezar a utilizar Unity. <p>Descargar Unity Hub</p> <ol style="list-style-type: none">1. Seleccionar <i>Descargar para Windows o Descargar para Mac</i> dependiendo del sistema operativo de su dispositivo.2. Una vez realizado se realizará la descarga, cuando termine de descargar, proceder a dar click en el archivo descargado. Esto lo guiará por los pasos para terminar la instalación de <i>Unity Hub</i>.
-------	--

	<ol style="list-style-type: none"> 3. Teniendo <i>Unity Hub</i> hay que dar de alta la licencia. Para esto seleccionar el engranaje en la parte superior y navegar al apartado de <i>License Manager</i>. 4. Seleccionar <i>Activate New License</i> y dar de alta una cuenta personal. <p>Descargar GUI Unity</p> <ol style="list-style-type: none"> 1. Dentro de <i>Unity Hub</i> navegar al apartado de <i>Installs</i>. 2. Seleccionar el botón de <i>ADD</i>. 3. Seleccionar la versión recomendada <i>Recommended Release</i> y dar <i>Next</i>. 4. Seleccionar <i>Visual Studio</i> y dar <i>Next</i>. 5. Con esto comenzará la descarga del GUI de <i>Unity</i> y una vez terminado ya se podrán correr los proyectos en este.
Repositorio	<ol style="list-style-type: none"> 1. Abrir el repositorio de GitHub del Proyecto aquí. 2. Presionar el botón verde que dice <i>Código</i>. 3. Copiar el enlace https presentado. (https://github.com/A00827463/movilidad_urbana.git) 4. Salir del navegador y abrir una nueva terminal(MacOS) o Command Prompt(Windows). 5. En la terminal/cp navegar al directorio en el que quieras guardar el repositorio de github dentro de tu dispositivo. 6. Para clonar el repositorio escribir en la terminal/cp: <code>\$ git clone https://github.com/A00827463/movilidad_urbana.git</code> 7. Esperar a que termine la descarga en la terminal, y una vez realizado esto, ya se tendrá acceso a la carpeta del proyecto dentro de su dispositivo.
Ejecución	<p>Para tener acceso al proyecto, primero hay que habilitarlo dentro de <i>Unity Hub</i>.</p> <ol style="list-style-type: none"> 1. Abrir la aplicación de <i>Unity Hub</i>. 2. Dar click en <i>ADD</i> para añadir un proyecto. 3. Dirigirte a la carpeta del repositorio que se clonó anteriormente, y elegir la carpeta dentro de este llamada <i>Movilidad Urbana</i>. 4. Teniendo esto, de regreso en <i>Unity Hub</i>, seleccionar el Proyecto de <i>Movilidad Urbana</i>. <p>Con esto ya tendremos acceso al proyecto y podremos acceder a la interfaz para modificar las escenas y los atributos dentro de la simulación.</p> <p>Para correr la simulación con los valores preestablecidos.</p> <ol style="list-style-type: none"> 1. En la parte superior de <i>Unity</i> se encuentran dos botones. <i>Play</i>, <i>Pause</i> y <i>Skip</i>. Seleccionar el botón de <i>Play</i> (Triángulo) y la simulación comienza. 2. Para terminar la simulación, presione el botón de <i>Play</i> nuevamente. <p>Para salir de Unity es tan fácil como cerrar la ventana de Unity dando click en la X de la esquina superior de la aplicación.</p>

Resumen

¿Por qué seleccionaron el modelo multiagentes utilizado? ¿Cuáles fueron las variables que se tomaron al momento de tomar la decisión? ¿Cuál es la interacción de esas variables con respecto al resultado de la simulación? ¿Por qué seleccionaron el diseño gráfico presentado? ¿Cuáles son las ventajas que encuentras en la solución final presentada? ¿Cuáles son las desventajas que existen en la solución presentada? ¿Qué modificaciones podrías hacer para reducir o eliminar las desventajas mencionadas?

Para el desarrollo de la solución de esta situación problema, decidimos generar nuestras entidades o agentes desde el mismo motor de simulación “Unity”, siendo que así sería mucho más sencilla la tarea de hacer que éstos operen de manera adecuada dentro de la simulación. Esto, debido a que ya contando con los agentes dentro del motor de simulación, los atributos o métodos específicos se les pueden agregar simplemente insertando el código que corresponda al agente de manera manual.

Los agentes que se tuvieron en consideración para el desarrollo del reto fueron: semáforos y automóviles. La interacción que se tiene entre estos dos dentro del motor de simulación puede darse de alguna de las siguientes maneras:

- Semáforo | Semáforo: cuando un semáforo de una determinada intersección cambia de estado (verde, amarillo o rojo) antes de que su contador se lo indique por alguna condición de congestión de tráfico, se debe de hacer que los demás semáforos también cambien de estado para así evitar colisiones.
- Semáforo | Automóvil: cuando un semáforo se encuentra en estado rojo, esto prohíbe al automóvil pasar por la intersección, y cuando el estado se encuentre en verde o amarillo, el automóvil tendrá la posibilidad de realizar el cruce. Además, cuando un semáforo detecte un automóvil y en los demás no se detecte alguno, dicho semáforo, en caso de encontrarse en estado rojo, cambiará a estado verde para permitirle al coche transitar.
- Automóvil | Automóvil: en caso de que un automóvil llegue a detectar otro coche dentro de un determinado rango, el primero se deberá de detener para evitar algún tipo de colisión.

Con respecto al diseño gráfico implementado dentro de la solución del reto, decidimos utilizar prefabs que se pueden encontrar en la Asset Store de Unity, siendo que esto nos permite ahorrarnos una gran cantidad de tiempo que iría destinada a elaborar el diseño del vehículo, texturas, etc.

Considerando la solución que como equipo decidimos abordar, dado que hacemos uso tanto de sensores como de un temporizador en cada vehículo, creemos que existe una gran ventaja con respecto a la reducción del congestionamiento vehicular, dado a que el sensor hace que las secciones donde realmente exista tráfico tengan la posibilidad de circular, mientras que aquellas donde no existan coches se mantendrán con un estado rojo. Sin embargo, también debemos de considerar que una desventaja que puede llegar a presentarse en un escenario real, es que esto mismo puede llegar a generar más congestionamiento en ciertas áreas. Esto debido a que muchos automóviles pueden llegar a bloquear completamente una calle, tapando inclusive la intersección, prohibiendo el paso a otros coches. Para este tipo de problemas, lo que se puede hacer para solucionarlo es implementar una sincronización entre todos los semáforos de una calle, ya sea de forma horizontal o vertical. Con dicha implementación, lo que se lograría es despejar completamente una gran cantidad de calles, reduciendo la posibilidad de estancamientos de tráfico.

Reflexión

Una reflexión sobre tu proceso de aprendizaje. Para ello, revisa el documento original que contiene tus expectativas al inicio del bloque y compáralo con las experiencias que viviste a lo largo de estas semanas.

Sebastian

Al realizar el reto de Movilidad Urbana, realmente pude poner a prueba mis actitudes de diseño y de programación en C#. La verdad que fue muy retador el poder hacer una simulación de esta magnitud al mismo tiempo que vamos aprendiendo cómo manipular los materiales, interfaces, y agentes en la clase. Sin embargo, con cada avance que estamos haciendo como equipo, sentía que las clases sí eran de ayuda. Ahora las clases fueron más de apoyo para entender los conceptos más que nada. Pero con el reto fuimos capaces de implementar estas funciones que nos dan a entender de una mejor manera la implementación de Unity y la implementación de Múltiples agentes. Me gusto mucho saber implementar assets en Unity, no solo porque hace la vida más fácil, sino porque hay muchos tantos que te permiten aprender diferentes conceptos, como el manejo de la cámara, manejo de objetos, etc. De cierta manera sí se cumplieron mis

expectativas, logramos implementar diferentes conceptos dentro de la interfaz de Unity. Sin embargo, siento que no fui capaz de aprender tantas cosas nuevas aparte de la implementación de multiagentes. Me quede queriendo mas y mas, hacer algo en conjunto, asombroso. Dicho esto, me voy satisfecho por todo lo que fui capaz de implementar en este reto.

Gabriel

Considero que realmente tuve un buen aprendizaje, siendo que el sector que más me interesaba, el cual era el desarrollo de código en C#, fue en el que realmente sentí que tuve un avance significativo. Esto debido a que gran parte de mi aportación para el desarrollo del reto se basó justamente en desarrollar código para algunas funcionalidades del proyecto final. Además, logré aprender distintos conceptos que pueden ser de ayuda para mi vida futura, como lo son los multiagentes, transformaciones, modelado, etc. Con respecto a estos conceptos previamente mencionados, considero que las transformaciones fue lo que más impacto tuvo en mí y en mi aprendizaje. Esto debido a que es algo que implementé durante el desarrollo de código en C#, y se vincula con la programación orientada a objetos en videojuegos, lo cual es un área más especializada que me sirvió para identificar que es un área en la cual posiblemente me gustaría desarrollarme profesionalmente.

Fernando

El proyecto fue desarrollado considerando distintos aspectos técnicos, lo que permitió adquirir aprendizajes variados y no centralizados. Fue tan diverso el proyecto, que trabajé desarrollo de código en c#, modelación con UNITY, interacciones entre agentes en Python y al mismo tiempo realicé pruebas y análisis de casos, aplicando métodos cuantitativos que permiten visualizar de manera gráfica los resultados. Para ser sincero, no esperaba un reto de esta complejidad, pero me dio la oportunidad de crecer en conocimientos que culminan en el desarrollo profesional que espero lograr durante mi carrera. El proyecto además me permitió comprender la complejidad de un entorno multiagente y al mismo tiempo entender la importancia de la representación digital de este tipo de entornos, ya que sin lugar a dudas son las simulaciones que más parecido tienen a la realidad. De este curso me llevo todos los conocimientos brindados por los profesores, además de los adquiridos por el desarrollo del trabajo práctico.