



Reto: Movilidad Urbana

Modelación de sistemas multiagentes con gráficas computacionales

Materia TC2008B

Integrantes del equipo

Martín Palomares García A01066569

Ana Fernanda Hernández Tovar
A01411484

Brian Alberto Salomón Sevilla
A00828826

Carlos G. del Rosal A01566719

Profesores

Luis Andres Castillo Hernández

Jorge Mario Cruz Duarte

Fecha de entrega



2 de Diciembre de 2021

Introducción

Partiendo del reconocimiento de la situación problema descrita. Posteriormente se genera una visualización de un sistema multiagente con gráficas computacionales que busca replicar este fenómeno humano con la finalidad de permitir análisis simple planteando dichas soluciones. Por ejemplo: Se implementan semáforos

inteligentes que brindan un flujo con mayor continuidad en un cruce de dos avenidas. Se mejoraría la actividad pasada por cruces viales de vehículos y podría traer enormes beneficios ambientales, económicos y sociales.

Descripción de Situación Problema

La movilidad urbana, se define como la habilidad de transportarse de un lugar a otro y es fundamental para el desarrollo económico y social y la calidad de vida de los habitantes de una ciudad. Desde hace un tiempo, asociar la movilidad con el uso del automóvil ha sido un signo distintivo de progreso. Sin embargo, esta asociación ya no es posible hoy. El crecimiento y uso indiscriminado del automóvil genera efectos negativos enormes en los niveles económico, ambiental y social en México.



Durante las últimas décadas, ha existido una tendencia alarmante de un incremento en el uso de automóviles en México. Los Kilómetros-Auto Recorridos se han triplicado, de 106 millones en 1990, a 339 millones en 2010. Ésto se correlaciona simultáneamente con un incremento en los impactos negativos asociados a los autos, como el smog, accidentes, enfermedades y congestión vehicular.

Para que México pueda estar entre las economías más grandes del mundo, es necesario mejorar la movilidad en sus ciudades, lo que es crítico para las actividades económicas y la calidad de vida de millones de personas.

Herramientas Utilizadas y sus Procesos de Instalación

La solución del reto se lleva a cabo sobre dos plataformas de desarrollo de software y gráficos computacionales: Unity y Python (utilizando Google Colaboratory).



Unity



El aprendizaje que abarcamos con Unity, comenzó con la modelación en 3D de la solución del reto. Como equipo, adquirimos conocimiento de una herramienta llamada ProBuilder para después utilizarla como el híbrido de modelado 3D con herramientas de diseño en niveles que es.

Como pueden ver en la imagen 1.1, se puede observar que el modelo está optimizado para partir de geometría simple pero es capaz de lograr edición detallada y la desenvoltura de UV (texturas).

Con esta herramienta, aprendimos a crear rápidamente prototipos de estructuras, características de una calle, vehículos, semáforos, etc, para lograr la modelación de la solución del reto.



Imagen 1.1 Simulación en Unity de la solución del reto

Los 3 conceptos base para la actividad:



Materiales: Son los componentes para que los elementos tengan un color, variando no sólo en matices, sino en opacidad, característica metálica o texturas.



Texturas: Con este elemento, se permite el revestimiento para un componente a través de un material, proporciona no solo diseños cargados desde imágenes, sino también texturas tridimensionales que pueden tener una interacción con el ambiente por sí mismas.



Iluminación: Este componente proviene de fuentes generales o puntuales, o bien, es aplicado indiscriminadamente sobre una sección u objeto en el ambiente. Este modifica los colores que son efectivamente mostrados gráficamente.

Proceso de instalación de Unity

El proceso de Instalación de Unity junto con ProBuilder es el siguiente:

1. Ingresa a la página de descarga
<https://unity3d.com/es/get-unity/download> y te aparecerán las siguientes opciones de descarga.
Es importante considerar que durante el proceso se pedirá una cuenta de Unity, que se puede crear gratuitamente con un correo electrónico.
2. Selecciona la opción **Unity Hub**
3. Después de ejecutarlo, te aparecerá lo siguiente
4. En la parte izquierda, encontrarás 3 botones, que son **Projects**, **Learn** e **Installs**. Le das click en “Installs” y le das click al botón de “**ADD**” en la parte superior derecha de la pantalla de Unity Hub para añadir una nueva versión.

Descargar Unity

Bienvenido! Está aquí porque desea descargar Unity, la plataforma de desarrollo más popular del mundo para crear juegos multiplatforma y experiencias interactivas 2D y 3D.

Antes de descargar, elija la versión de Unity que sea adecuada para usted.

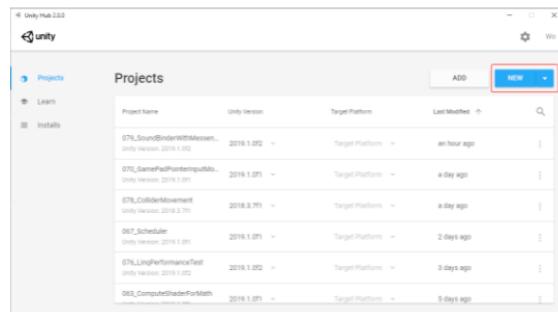
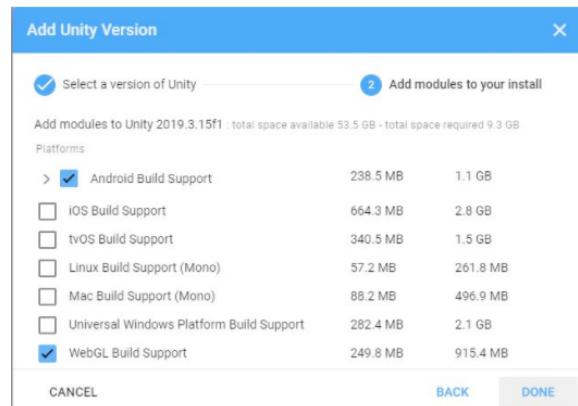
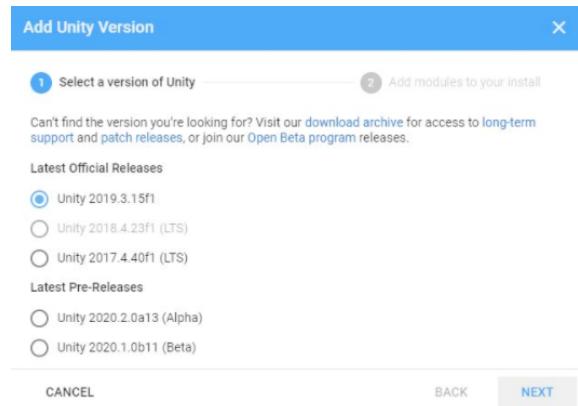
[Elige tu Unity + descargar](#) [Descarga Unity Hub](#)

[Descubrir más acerca del nuevo Unity Hub aquí.](#)

The screenshot shows the Unity Hub interface. On the left, there are three tabs: 'Projects' (selected), 'Learn', and 'Installs'. Under 'Installs', there is a table with the following data:

Project Name	Unity Version	Target Platform	Last Modified
079_SoundBinderWithAssets	2019.1.0f2	Target Platform	an hour ago
070_GamerPadInputUpdate	2019.1.0f1	Target Platform	a day ago
078_ColliderMovement	2018.3.7f1	Target Platform	a day ago
067_Scheduler	2019.1.0f1	Target Platform	2 days ago
076_LingPerformanceTest	2019.1.0f2	Target Platform	3 days ago
063_ComputeShaderFunMath	2019.1.0f1	Target Platform	5 days ago

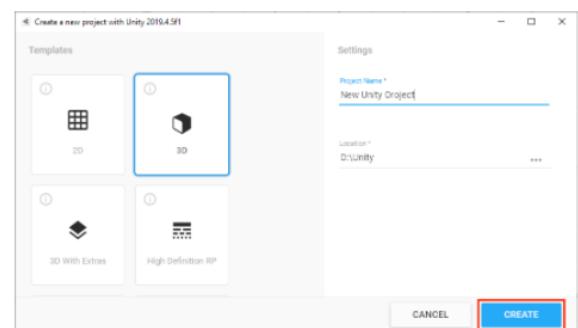
5. Escoge la versión que te interese, y le das click a next (Las que están en gris es ya están instaladas)
6. Ahora, escoge las plataformas sobre las que vamos a generar el juego. Después de seleccionar los elementos seleccionados, das click en el botón DONE y se inicia la descarga e instalación de los elementos escogidos.
7. Una vez instalada podemos ver que plataformas soporta y podremos crear nuevos proyectos en la sección Projects.



Instalación de ProBuilder

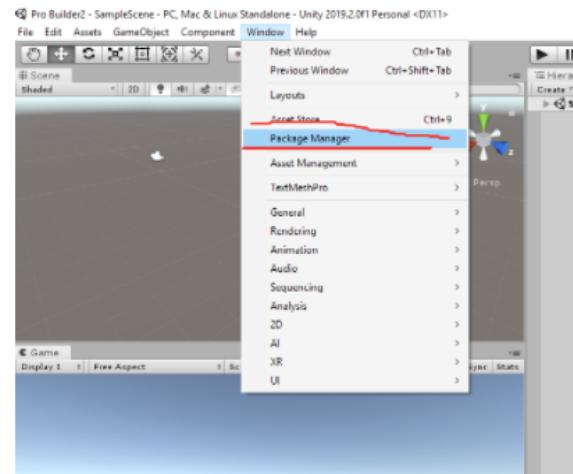


Creas tu proyecto 3D en Unity

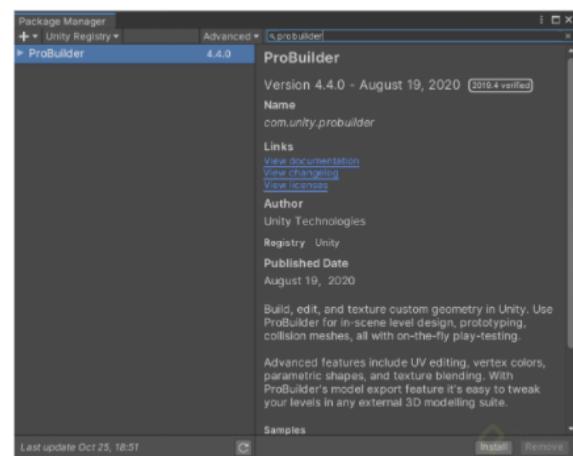




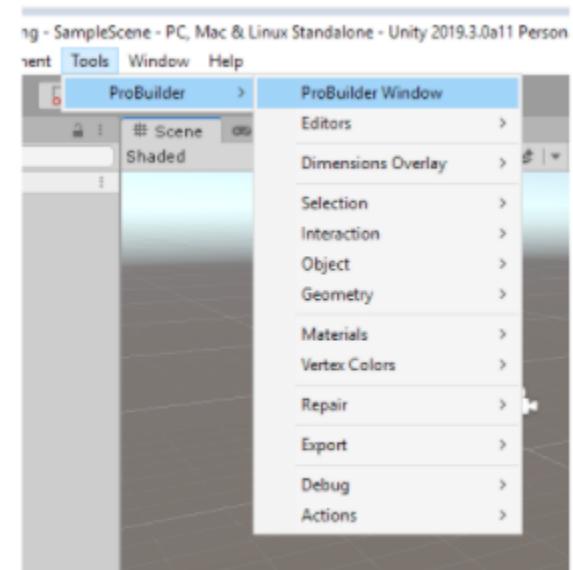
Después de que se haya generado el proyecto, le das click a “**Window**” y luego le das click a “**Package Manager**”



Después aparecerá la siguiente pantalla, y busca en el buscador “**Probuilder**” y le das click en el botón de “**Install**” en la esquina inferior derecha.



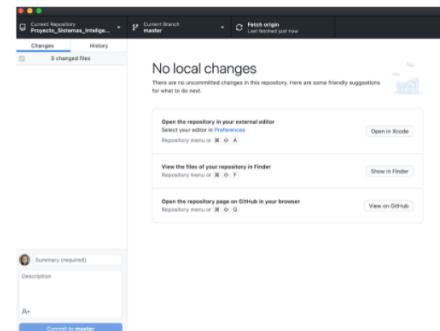
Ya cuando termine de instalarse, en tu barra, ve a “**Tools**” y da click en “**ProBuilder**” y después das click en “**ProBuilder Window**” y te aparecerá la barra de herramientas de ProBuilder.



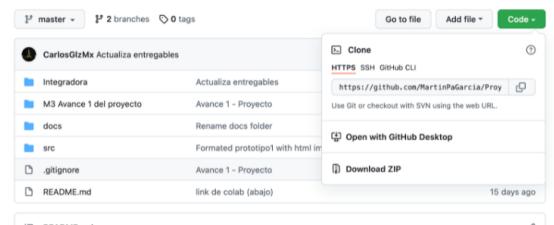
Con Unity y Probuilder correctamente instalados, es posible ejecutar el proyecto a través de la importación de archivo “.unitypackage” entregado por el equipo. Si se desea contar con la distribución completa del proyecto, es posible clonarlo desde Github a través de los siguientes pasos:



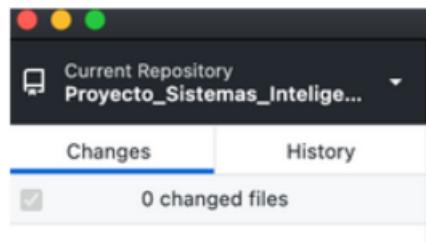
Descargar la aplicación GitHub Desktop del siguiente link:
<https://desktop.github.com/>



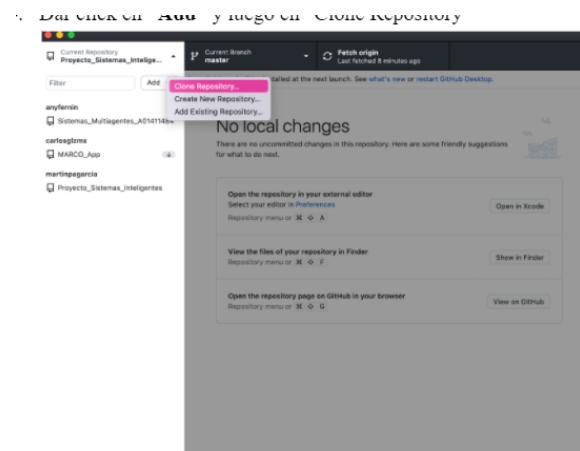
Ir al link de la carpeta de Github (Se encuentra en la sección de “**Ligas Relevantes del Proyecto**”) y copiar el link de “Clonar repositorio”



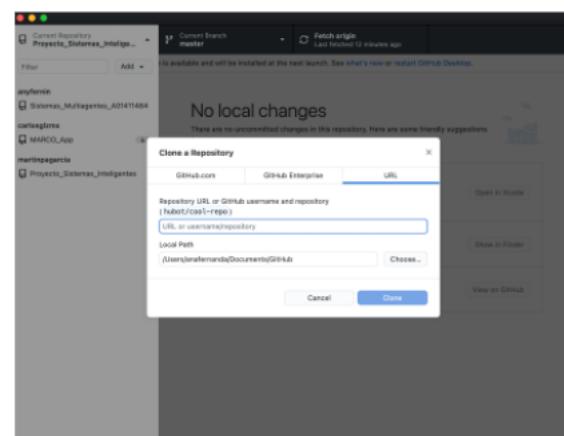
Ir a Github Desktop y dar click en Current Repository



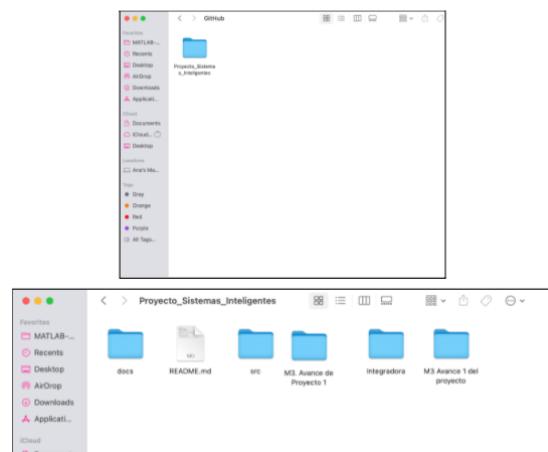
Dar click en “Add” y luego en “Clone Repository”



Dirigirte al apartado de URL, copiar el link del repositorio y dar click en “Clonar”



En tu ordenador, buscar el Local Path (donde tienes la carpeta del repositorio) y encontrarás la carpeta del proyecto junto con los archivos que están en el Github.



Python

Para hacer la simulación en 2D de la solución de la situación problema, utilizamos la herramienta de Google Colaboratory, en la cual programamos con Python. Las distintas actividades del curso que llevamos a cabo de esta misma manera se presentaron como una construcción incremental, donde un porcentaje alto de los códigos anteriores fue de utilidad para los códigos en el momento. Es entonces que la implementación actual no es más que una integración de conocimientos existentes.



Es entonces que la implementación actual no es más que una integración de conocimientos existentes. El sistema multiagentes es ahora capaz de manejar semáforos inteligentes, vueltas y simulaciones de larga duración.

Una percepción de los trabajos que se han hecho se puede obtener viendo las librerías que se utilizan en la actividad de Python para este problema:

Librerías que se utilizan en la actividad de Python para este problema:



Mesa: Una herramienta para modelar sistemas multiagentes basados en dos clases esenciales Model y Agent, que trabajan al ambiente con reglas y estados y a los agentes con voluntades y estados. Multigrid permite contar con una cuadrilla que almacene tantos agentes sea necesario en un solo espacio, SimultaneousActivation sirve para que se pueda activar a todos los agentes en un mismo calendario y DataCollector sirve para recolectar en una variable los estados instantáneos del ambiente.



Pandas y Numpy: Módulos matemáticos también fundamentales en la industria que en este caso proporcionan una fácil construcción y manipulación de matrices conteniendo información sobre el ambiente y las animaciones.



Random: Nativo de Python que provee números pseudoaleatorios útiles para la generación siempre distinta de vehículos desde cualquier dirección.



Matplotlib: Módulo esencial para gráficos y en nuestro caso, animaciones. No solo permite visualizar el sistema multiagentes, sino también personalizar las formas, colores y tamaños con los que se muestran estos datos.



Ngrok: Es una herramienta de proxy inverso que abre túneles seguros desde URL públicas hasta localhost, perfecta para exponer servidores web locales, crear integraciones de webhook, etc. Permite la conexión con Unity simulando que el servidor y la modelación en 3D son componentes independientes en dispositivos distintos.



Json: Es un formato de datos liviano con muchas similitudes con los diccionarios de Python que permite transmitir información a otros lenguajes de programación o herramientas computacionales, por ejemplo, las instrucciones del sistema a Unity.



Python: Es un formato de datos liviano con muchas similitudes con los diccionarios de Python que permite transmitir información a otros lenguajes de programación o herramientas computacionales, por ejemplo, las instrucciones del sistema a Unity.

Simulación

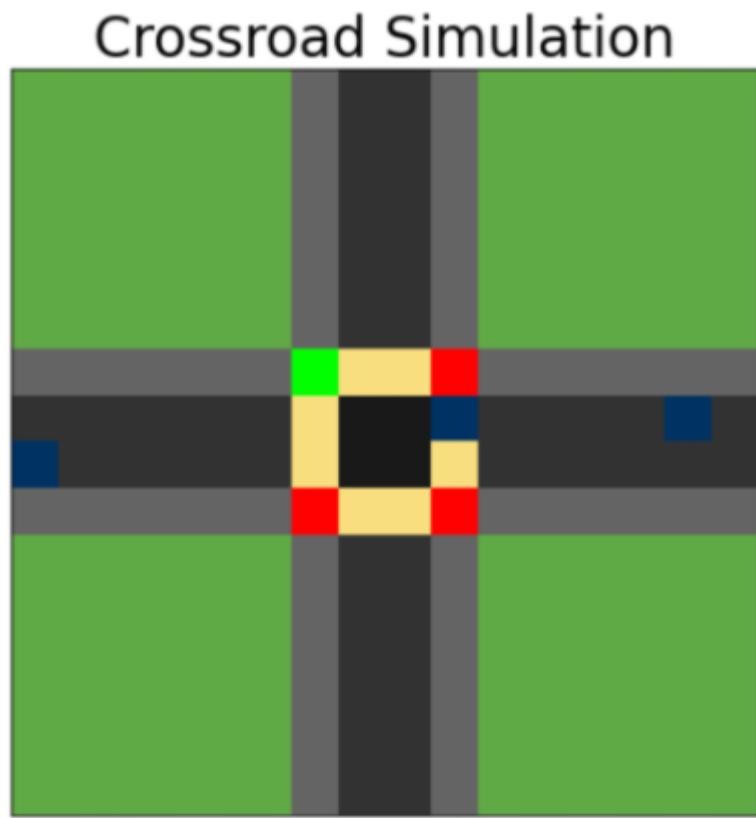
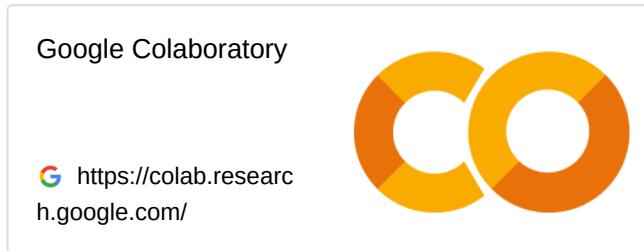


Imagen 1.2 Simulación en Google Colaboratory del Reto

Para esta herramienta, no se requiere de instalación, solamente con ingresar al siguiente link, puedes crear un nuevo proyecto en Google Colaboratory.



Se entrega además, presente en el repositorio del equipo de trabajo, un programa de Python adaptado para la ejecución local del servidor. Conectando a Unity solamente con localhost, sin depender de ngrok, permite un número ilimitado de peticiones y una respuesta más rápida, lo que permite ejecutar la simulación con velocidades mayores sin incurrir en problemas.

Diagramas

Los diagramas que desarrollamos para entender el funcionamiento y la creación del proyecto en simulación, son los siguientes:

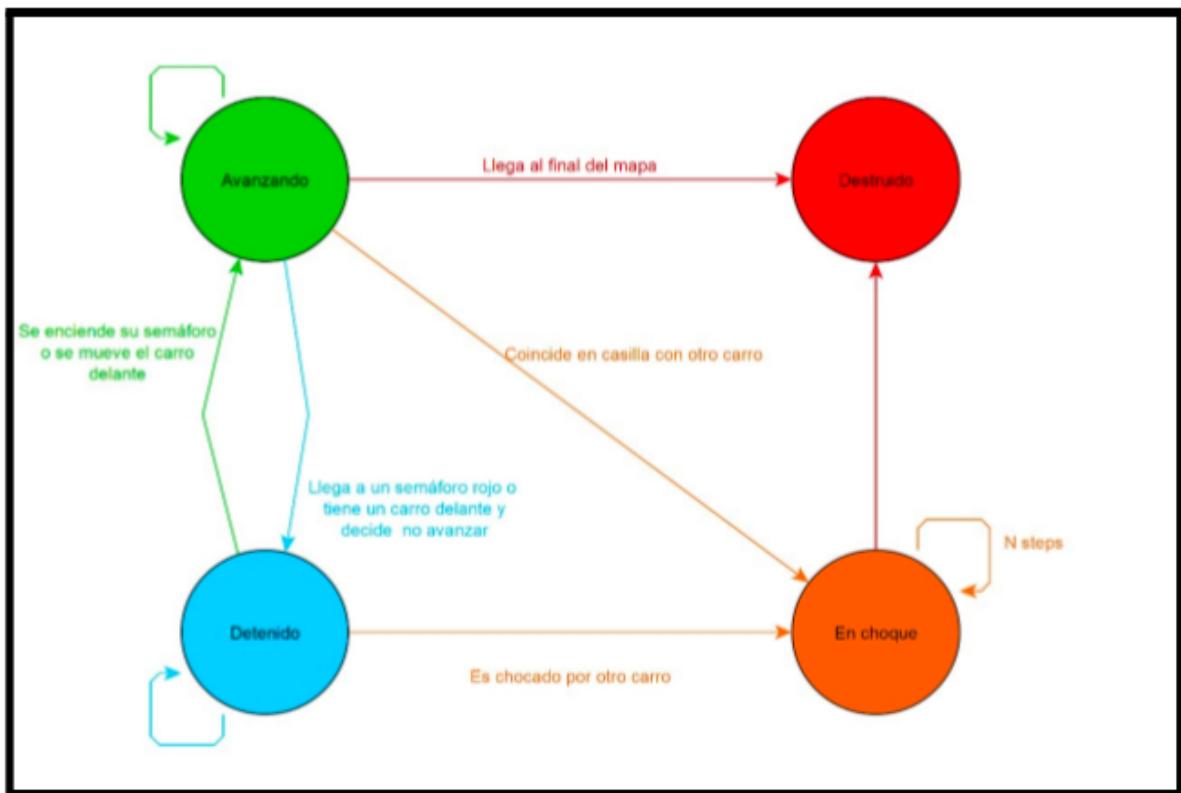


Imagen 1.3 Máquina de estados de los vehículos.

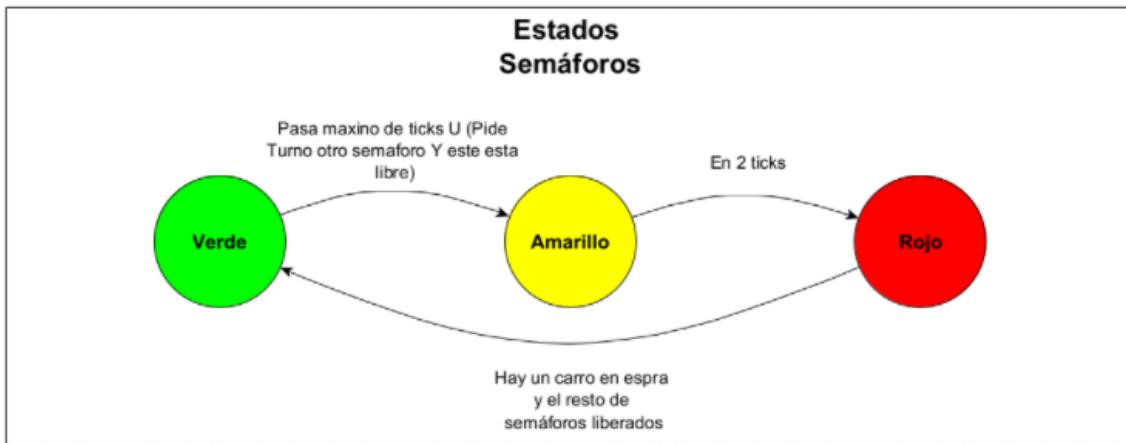


Imagen 1.4 Máquina de estados de los semáforos.

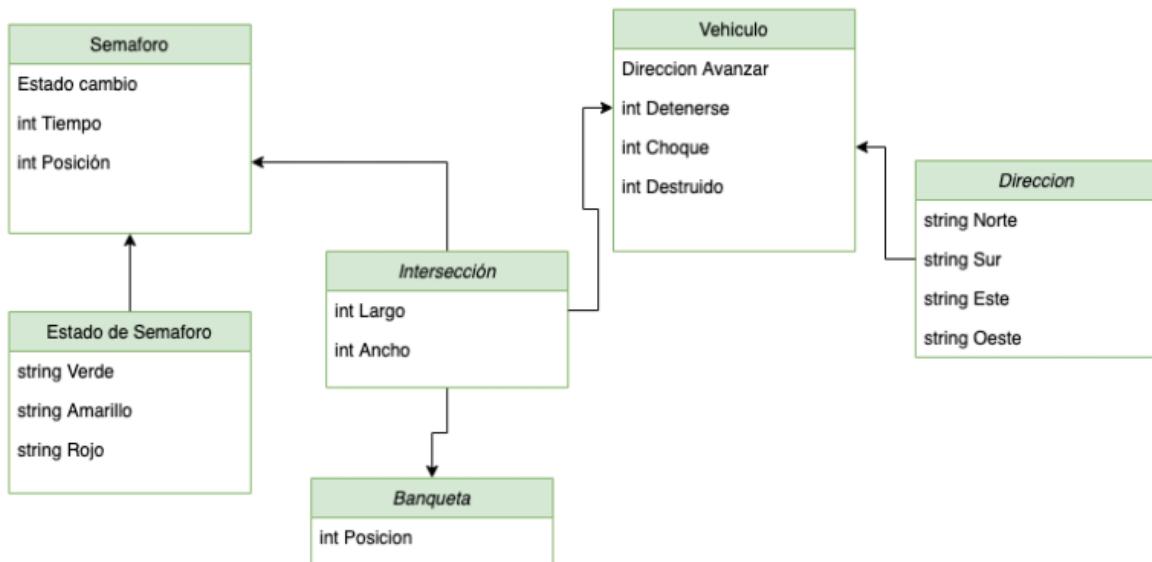


Imagen 1.5 Diagrama de Clases

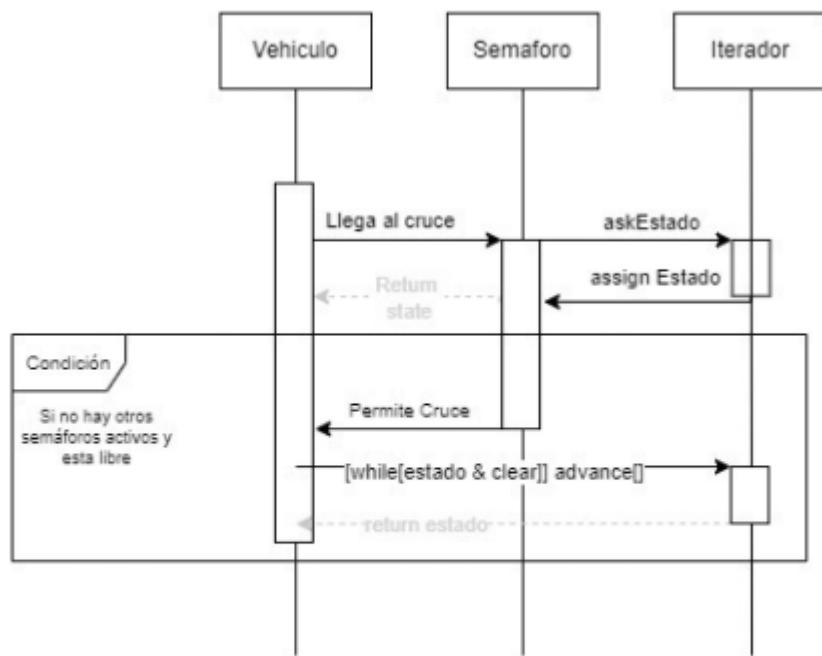


Imagen 1.6 Protocolo de Interacciones

Ejecución del proyecto

Una vez que se hayan importado todos los componentes del archivo “.unitypackage” entregado a un proyecto nuevo de Unity, o se haya clonado el proyecto completo desde Github, se debe de hacer una pequeña configuración previa. En primera instancia, para maximizar el renderizado de muchos componentes de iluminación utilizados (semáforos, iluminación de las calles, faros de los coches) es importante abrir **Edit > Project Settings > Quality** y seleccionar el perfil **High, Very High, o Ultra**, como perfil por defecto (Haciendo click en la sección default justo debajo). Esto permitirá que la simulación se ejecute con altos recursos y permita un renderizado de las texturas óptimo.

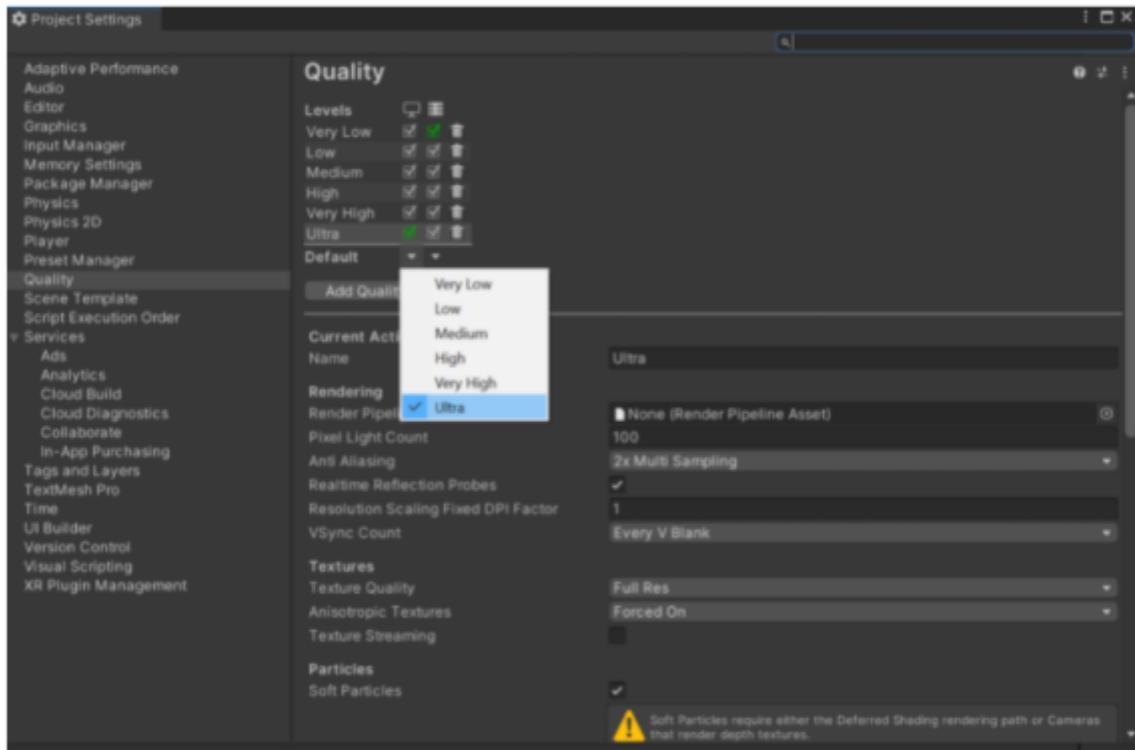


Imagen 1.7 Configuración de Calidad

Después, es necesario contar con la liga de conexión de Python. Si se ejecuta desde Google Colab en la liga proporcionada, se debe de copiar la liga pública de ngrok que se obtiene tras correr todas las celdas, aquella terminando en “.io”. En cambio, si se utiliza el archivo de Python en el mismo ordenador, se debe de copiar la liga de “localhost”. Cual sea el caso, esta liga se debe de copiar en el espacio URL del inspector cuando se selecciona el componente de Unity llamado “Simulation”. (Nota: Si se trabaja con Colab y Ngrok esta última herramienta cuenta con un límite de peticiones por minuto, por lo que el parámetro de “Step Duration” debe de quedar en un mínimo de 2 para evitar que se corte la comunicación).

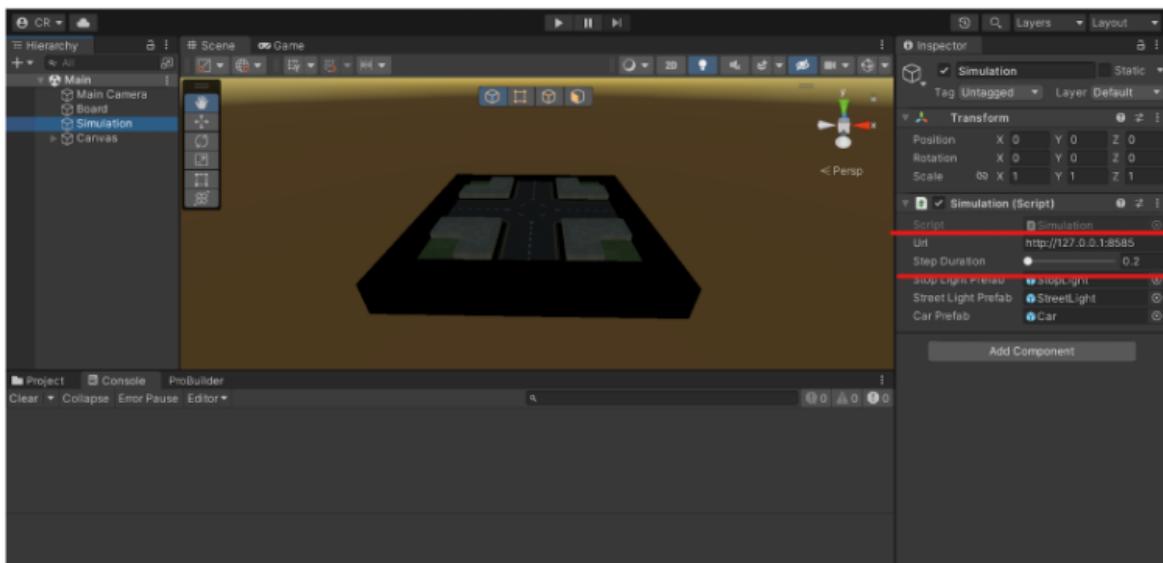


Imagen 1.7 Configuración en el Inspector de la Conexión

Es entonces que, sin haber interrumpido la ejecución de Python, se debe de correr la escena en Unity. Esta continuará hasta alcanzarse el tiempo máximo de ejecución elegido en Python o hasta que Python sea interrumpido a través del teclado (Ctrl + C). Este fin de la comunicación con Unity permitirá que en Python se muestre la animación generada en 2D con matplotlib.

Conclusión

En conclusión, aprender acerca de la problemática de este reto, nos ayudó a comprender como equipo sobre la situación que está presente en nuestro país, pero con el uso de la tecnología, podemos implementar una solución viable para este tipo de problemas, y no solo para este problema, sino para problemas futuros que se irán presentando. Además, podemos observar todo el valor que proporciona el saber identificar los agentes, modelos y trabajos por realizar en el reto, al igual que ir construyendo poco a poco la entrega de esta actividad gracias a los conocimientos adquiridos en clases. Vemos también los detalles que pueden mejorarse para entregar cada vez un trabajo más detallado, avanzado y con posibilidades de escalar a objetivos y aprendizajes mayores. Crear agentes inteligentes, resolver problemáticas actuales e integrar con visualizaciones de Unity son tareas que ahora están al alcance del equipo de trabajo.

FAQ



¿Por qué cuando corro el código de python, en google colab, me aparecen errores en el código diciendo que no reconoce a los agentes y/o modelos?

Esto se debe principalmente a que no se han cargado las librerías en el código. Para que el programa se ejecute sin problemas, primero se tienen que ejecutar las librerías de python que estamos llamando al principio del programa. En el caso de cualquier comportamiento inesperado, es mejor ejecutar la totalidad de las celdas en Colab, o bien, el programa completo de Python de nuevo.



¿Por qué Unity muestra los componentes de color rosado sin sus texturas observadas en las imágenes que proporciona el equipo?

Esto se debe principalmente a que no se han cargado las librerías en el código. Para que el programa se ejecute sin problemas, primero se tienen que ejecutar las librerías de python que estamos llamando al principio del programa. En el caso de cualquier comportamiento inesperado, es mejor ejecutar la totalidad de las celdas en Colab, o bien, el programa completo de Python de nuevo.



¿Por qué no sucede nada al correr la escena de Unity?

Esto sucede debido a que no hay una conexión correcta con la simulación. Verifique que el programa de Python no haya sido interrumpido, que la liga ingresada en Unity sea la liga más reciente generada, y que de estar trabajando con Colab, el ordenador cuente con una conexión de internet que no limite este tipo de comunicaciones. Por ejemplo, en las instalaciones del Tec de Monterrey, la infraestructura de seguridad siempre bloquea la conexión de Unity con Python por medio de Ngrok.

Ligas Relevantes del Proyecto

En esta sección se encuentran los links al repositorio de trabajo, la herramienta de comunicación y las libretas de Google Colaboratory:

Github

GitHub - MartinPaGarcia/Proyecto_Sistemas_Inteligentes

Martín Palomares García A01066569 Brian Alberto Salomón
Sevilla A00828826 Ana Fernanda Hernández Tovar A01411484
Carlos González del Rosal A01566719 La comunicación por


https://github.com/MartinPaGarcia/Proyecto_Sistemas_Inteligentes

3 Contributors 0 Issues 0 Stars 0 Forks

Slack Group:

Slack

 https://join.slack.com/t/itesm-zfs6300/shared_invite/zt-yekqz1xf-YB6Vvd8T_5uPp0oohGWOvQ

Google Colab:

Google Colaboratory


<https://colab.research.google.com/>

Para simulación multiagentes:

Google Colaboratory


<https://colab.research.google.com/drive/1qRDmYQBwF1zbWBVqYhZgUxfNukl5D0v?usp=sharing>

Con un servidor abierto a Unity:

Google Colaboratory

 <https://colab.research.google.com/drive/1d0f4LAI-MwrjgPfyw4cBjk-gaLIWV4IO?usp=sharing>

