

Actividad integradora

Noviembre 24 2021

Descripción del problema

Esta es una actividad individual en la que vas a integrar todo lo aprendido hasta el momento. Se trata de una versión simplificada del reto que se evaluará individualmente:

- Las calles pueden tener solo dos carriles
- Todos los modelos pueden ser objetos simples, como cajas, y esferas

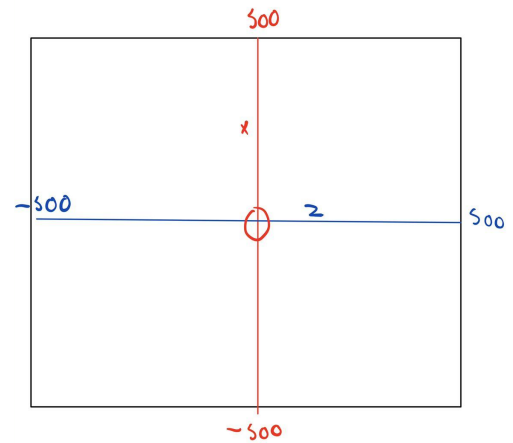
Parte 1. Sistemas multiagentes

Considera la situación en la que se tiene un paso peatonal en una avenida de dos carriles, para el cual hay un semáforo que ayuda a los peatones a detener el tráfico. Dicho semáforo se pone en rojo por al menos 60 segundos, mientras que la luz en verde permanece encendida hasta el momento en que algún peatón presiona el botón que solicita la luz roja. Para avisar a los conductores que el semáforo está por cambiar a rojo, la luz amarilla aparece por 5 segundos.

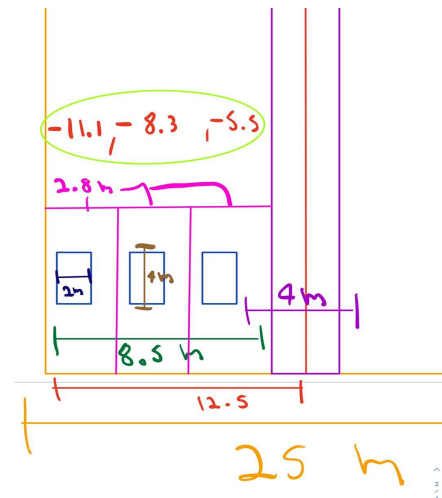
Para esta situación:

1. Describa cómo representarías el entorno en una retícula rectangular de $N \times M$ casillas.
 - En este caso decidimos basarnos en la vida real tomando como referencia el cruce de Aviación - Ramón corona, usamos la herramienta de Google maps para tomar diversas medidas de las calles, los pabellones, el cruce y los semáforos, así como comenzamos a planear los edificios aledaños para modelarlos y dar un toque más realista.

- Con la herramienta de google maps tomamos distintas medidas del cruce, decidimos tomar el centro del cruce como el origen, así como decidimos modelar 500 metros a cada extremo del cruce, generando un rectángulo con medidas de 1000 x 1000, medimos que cada calle mide 25 metros de ancho y contiene un pabellón en medio, el pabellón varía en ancho por lo que decidimos usar la parte más ancha de alrededor de 4 metros como referencia



- Para poner cada agente calculamos en tres carriles donde se encuentra el centro de cada carro, estas coordenadas las reflejamos en el otro carril en el punto más abajo ya que como es el carril derecho los autos van de abajo para arriba.



2. Enliste las diferentes situaciones (percepciones del estado del entorno) a las que se enfrentarían los conductores.

Hay un semáforo para cada carril que puede tener tres estados verde, amarillo y rojo, la velocidad de cada agente se calcula según el carro de adelante y un factor estocástico, si la distancia entre agentes es de menos de dos metros existe un choque y se paran en su lugar (Por lo que paran el tráfico), la luz del semáforo indica si los carros comienzan a frenar (amarillo), pueden pasar (verde) o tienen que estar parados (rojo)

3. Defina las acciones que llevarían a cabo los conductores para cada una de las situaciones que consideraste en el punto anterior.

La única acción es la de cambiar de velocidad según lo explicado anteriormente:

“La velocidad de cada agente se calcula según el carro de adelante y un factor estocástico, si la distancia entre agentes es de menos de dos metros existe un choque y se paran en su lugar (Por lo que paran el tráfico), la luz del semáforo indica si los carros comienzan a frenar (amarillo), pueden pasar (verde) o tienen que estar parados (rojo)”

4. Programe una simulación en Python para esta situación. Recopila información tal como velocidad a la llegada del semáforo, cantidad de autos detenidos cuando está en rojo el semáforo, etc.

5. ¿Qué pasaría en la simulación si el tiempo en que aparece la luz amarilla se reduce a 0?

El frenado de los agentes cerca del cruce sería demasiado en seco ya que no tendrían tiempo para adaptarse al cambio de luz, incluso unos agentes se podrían pasar el alto.

Notas:

Para este problema, considera que el semáforo se pone en rojo cada cierto tiempo de manera aleatoria. De esta forma, no es necesario simular a los peatones e incluirlos en la simulación.

Parte 2. Gráficas Computacionales

Datos de la simulación

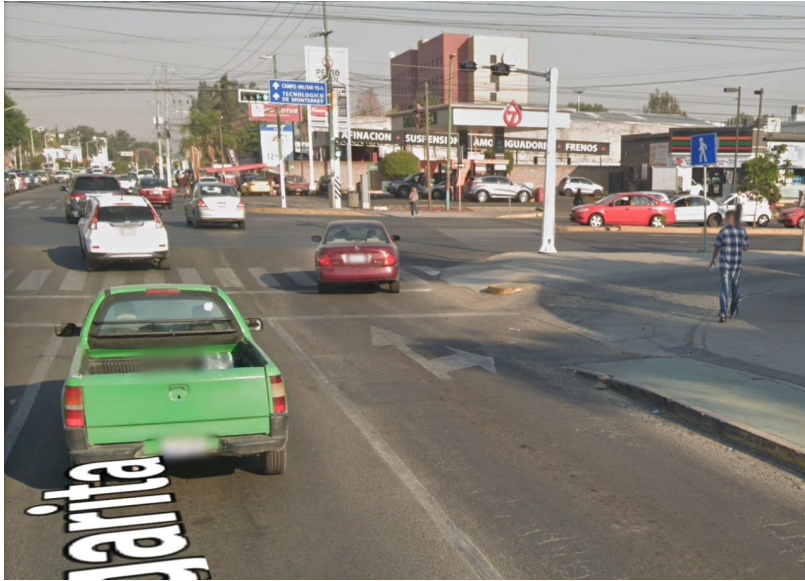
- Los datos son leídos de un objeto en formato JSON

Modelado

Los distintos modelos pueden ser simples (cajas, esferas) y no requiere incluir Nivel de Detalle (LOD):

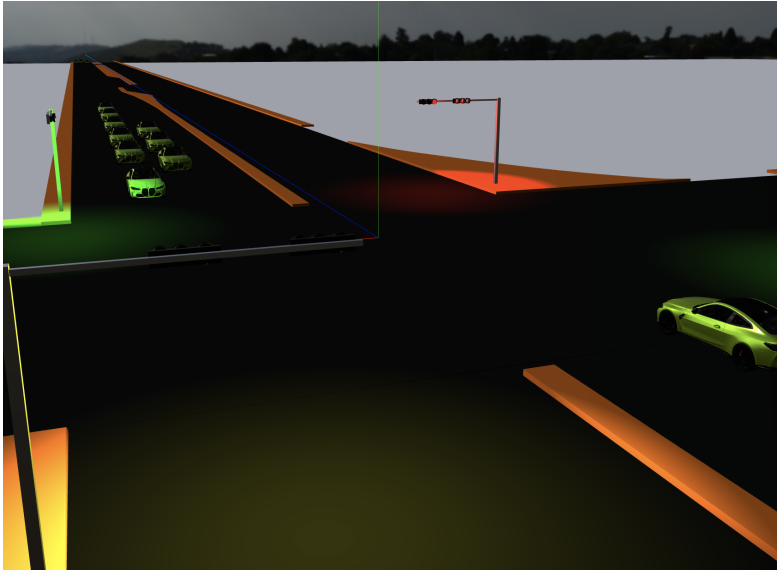
- Auto
- Semáforo
- Edificios
- Calles





Materiales

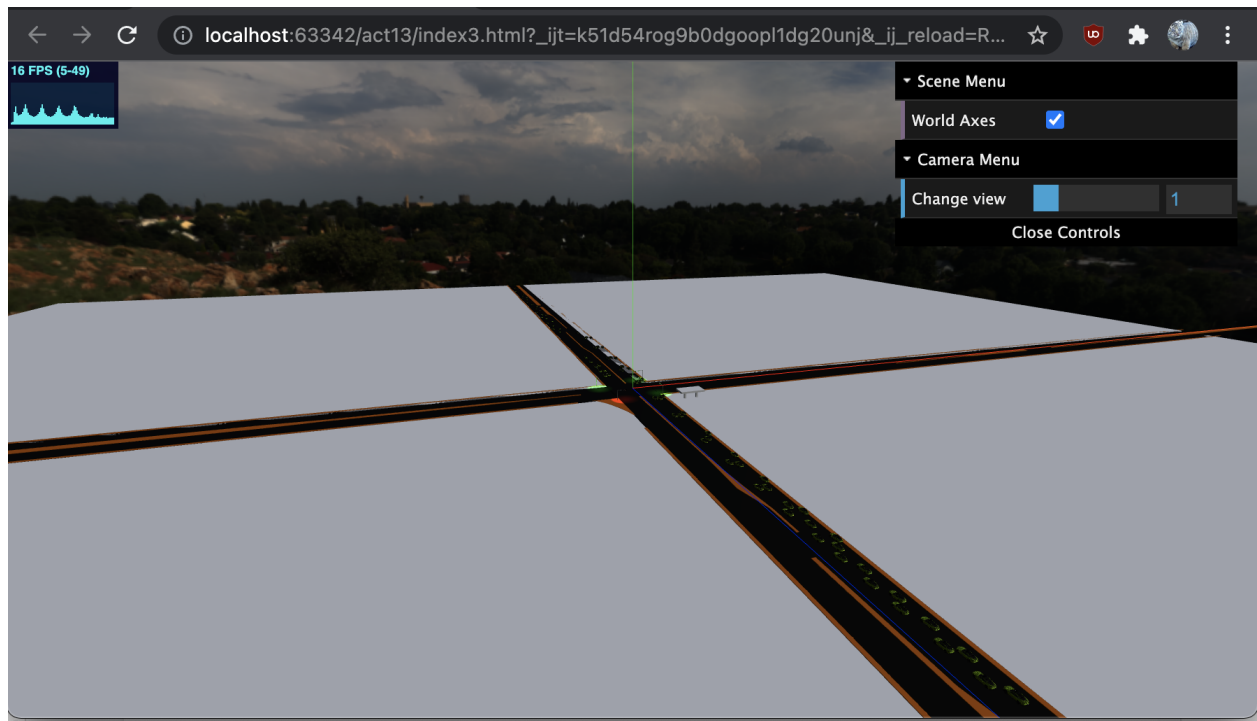
Los materiales de los modelos deben incluir las opciones de wireframe, colores y texturas:



Cámaras

Se deberá incluir varias vistas de la escena:

- Vista en perspectiva desde una posición fija
- Vista superior fija
- Vista desde el interior de un auto y que se moverá con el auto
- Opcionalmente, una vista orbital alrededor de un punto (intersección, o un auto)



Viewports

- El usuario podrá seleccionar las distintas vistas entre uno o más viewports

Animación

- Los automóviles deberán desplazarse sobre las calles y reaccionar a otros autos y al semáforo en la intersección



Iluminación (Opcional)

- Una fuente de luz ambiental
- Al menos una fuente de luz direccional.

- Opcionalmente, una fuente de luz puntual sobre cada auto que se moverá con el movimiento del auto.

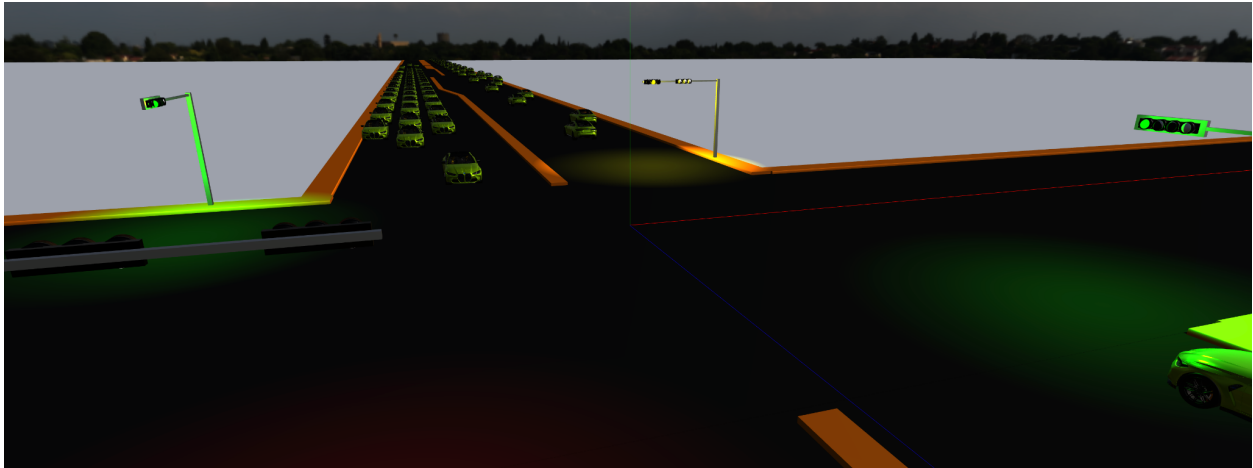
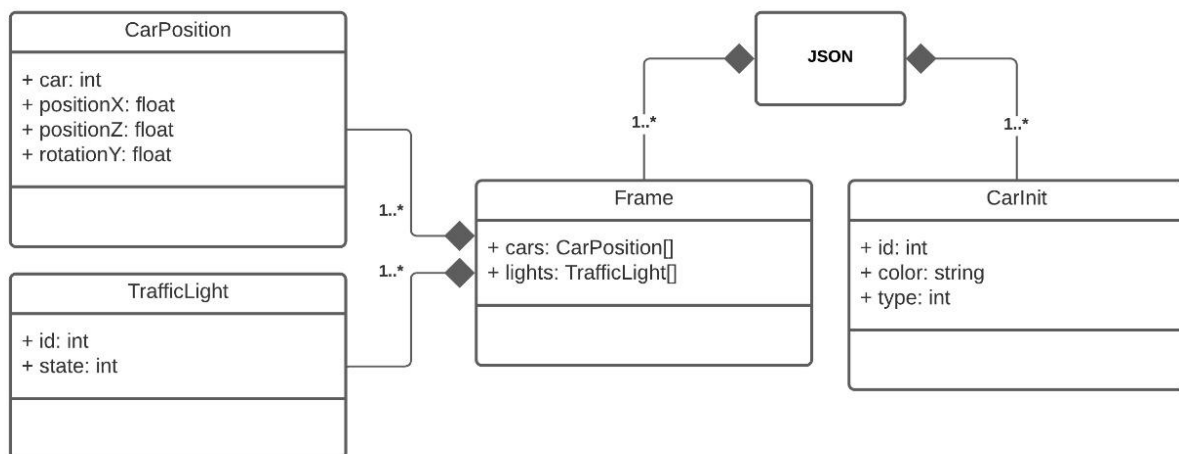


Diagrama de clases de comunicación JSON gráficas:



Entregas

Deberás entregar el enlace a un repositorio personal de GitHub que contenga: • Un documento PDF con los diagramas de clases, la descripción del entorno multiagente (entorno, percepciones, acciones).

- El código implementado para la simulación.
- El código de la App gráfica: archivo index.html (CDN version) y todos los archivos de textura y JSON necesarios para ejecutar la solución, incluyendo el archivo readme.

Rúbrica de evaluación

- Evaluación individual (preguntas abiertas): 30%
- Demostración de la funcionalidad: 40%
- Calidad y contenido del reporte: 15%
- Código completo: 15%