



TC2008B. Modelación de Sistemas Multiagentes con Gráficas Computacionales

Actividad Integradora

Lizbeth Maribel Melendez Delgado

A01232559

24 Noviembre del 2021

Tabla de contenidos

Descripción del problema	3
Parte 1. Sistemas multiagentes.....	3
Agentes y entorno	3
Percepciones del estado	4
Simulación	5
Parte 2. Gráficas computacionales	5
Código.....	5
Componentes.....	5
Archivo JSON.....	5
Autos	6
Escenario	6
Cámara	7
Luces	7

Descripción del problema

En una avenida de dos carriles (sentido contrario), se cuenta con un paso peatonal en el cual hay un semáforo que permite a los peatones detener el tráfico. Este semáforo se mantiene en verde hasta que el peatón presiona un botón, después pasa a amarillo por 5 segundos y luego a rojo donde tiene una duración de 60 segundos.



Imagen 1 Descripción del problema

Parte 1. Sistemas multiagentes

Agentes y entorno

El primer paso para realizar la modelación de multiagentes, fue identificar nuestro agentes. En este caso, se consideraron dos, el semáforo y automóvil, cada uno con las siguientes características:

Agentes	Semáforo	Auto
Propiedades constantes	Posición (x,z) Dirección	Dirección
Propiedades variables	Estado	Velocidad Posición (x,z) Estado
Acciones	Actualizar su estado (verde, amarillo o rojo)	Actualizar su estado (si esta en movimiento o detenido) Avanzar Modificar velocidad Detenerse

Tabla 1 Definición de agentes

Se optó por representar el entorno en un espacio continuo, utilizando la herramienta de agent py en Python. De igual forma, se podría representar con una matrícula de NxM, donde N sería dos representando los carriles de la avenida y M el numero de posiciones en las que puede estar un carro a lo largo de la avenida. Para conocer el entorno, los predicados del dominio que se tomaron en cuenta son los siguientes:

- esEstado(“verde/amarillo/rojo”)
- esDireccion(“norte/sur”)
- esCarroEnMovimiento()
- esDistanciaMin()

Percepciones del estado

Como se menciona en el punto anterior, los agentes en esta simulación son los conductores y el semáforo y sus acciones están sujetas a ciertas condiciones del entorno que determinan una situación. Como se realizo la modelación tomando los agente como reactivos, se definen las siguientes acciones, por medio de la forma situación → acción.

	Situaciones ¹	Acciones	Resultados
1	Si hay un auto delante	El auto se detiene	La velocidad del auto es 0
2	Si más adelante hay un auto	El auto frena	La velocidad del auto disminuye
3	Si el semáforo esta en amarillo y se encuentro cerca del semáforo	El auto acelera	La velocidad del auto aumenta
4	Si el semáforo está en amarillo y se encuentra lejos del semáforo	El auto frena	La velocidad del auto disminuye
5	Si el semáforo esta en rojo	El auto se detiene	La velocidad del auto es 0

Tabla 2 Definición de situación -> acción

Y su precedencia es la siguiente

$$1 < 2 < 3 < 4 < 5$$

En este caso se tiene cierta duración definida a cada estado del semáforo y es importante considerarlo en la definición de situación → acción del agente ya que, por ejemplo, en caso que el

¹ Percepciones del estado del entorno

tiempo de duración del semáforo amarillo fuera 0, el auto no tendría tiempo de frenar a tiempo y si se encuentra un carro muy cerca podría chocar.

Simulación

El código de la simulación hecho en Python se encuentra en el siguiente enlace: [simulación avenida](#).

Parte 2. Gráficas computacionales

Código

Después de realizar la simulación de multiagentes, se representa la información obtenida en una aplicación gráfica empleando HTML y Javascript, en específico la librería Three.js. El código de la app se encuentra en el siguiente enlace: [aplicación gráfica](#).

Componentes

Los componentes principales que se muestran en la aplicación y permiten la visualización de los resultados de la simulación de multiagentes son:

- Datos (resultados de modelado de multiagentes) en formato JSON
- Autos
- Escenario
- Cámara

La mayoría de los componentes que forman esta aplicación cuentan con un menú que permite al usuario modificar ciertos parámetros. A continuación se explica mas a detalle la implementación, funcionalidad y opciones de menú de cada uno de ellos.

Archivo JSON

En esta parte fue importante definir cuales datos eran necesarios guardar para representarlos en el espacio gráfico. Primero se tiene una lista que guarda otra lista de carros, semáforos y frames. La lista de carros y semáforos contiene el id, posición inicial (x, z) y dirección de todos estos objetos. Mientras que la lista de frames contiene a su vez otra lista de carros y semáforos, los carros cada uno con la posición(x, z) y el id, y los semáforos con su estado.

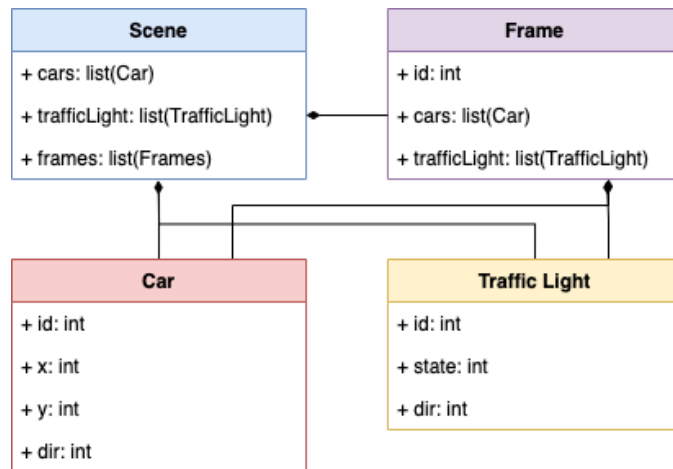


Imagen 2 Diagrama de datos

Autos

Para representar los automóviles en la aplicación se utiliza un modelo con formato obj, al cual se le agrega el material de wireframe y color.



Imagen 3 Modelo de automóvil

Además se cuenta con un menú que tiene las siguientes opciones:

- Hacer visible o invisible
- Mostrar/quitar/cambiar color del material wireframe
- Mostrar/quitar/cambiar color del material color

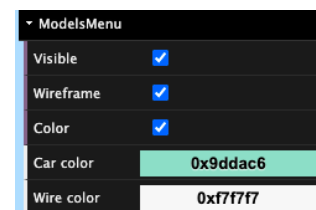


Imagen 4 Menú de modelo de auto

Escenario

El escenario esta compuesto por los ejes, la avenida de dos carriles, el cruce peatonal, los semáforos, banqueta y edificios. Para modelar estos componentes se utilizan diversos recursos que ofrece la librería de Three.js como grupos, mesh, formas geométricas, aplicación de texturas, entre otros. Cada uno de los elementos se implementa con una clase y al igual que el automóvil, se da la opción al usuario de modificar la visibilidad y material de los edificios que conforman el escenario.

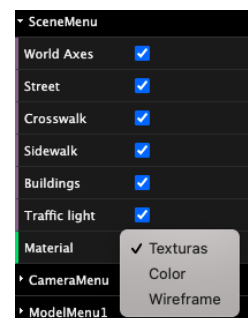


Imagen 5 Menú escenario

Por ejemplo, a continuación se muestran los resultados de los distintos materiales en los que se pueden visualizar los edificios.



Imagen 6 Material: textura (izquierda), color (centro) y wireframe (derecha)

Cámara

Para la cámara se tiene una clase, la cual tiene distintas propiedades y métodos que permiten al usuario ver el escenario desde distintas vistas como lo son: desde cierta perspectiva, de la parte superior y desde el interior de un auto.

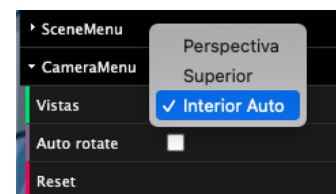


Imagen 7 Menú de cámara

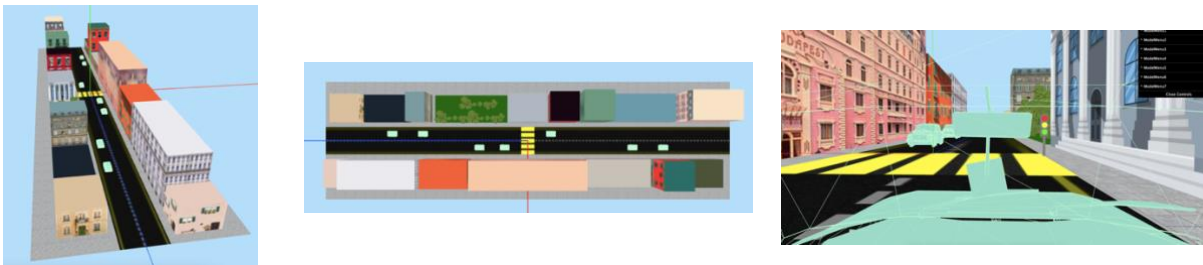


Imagen 8 Vistas: perspectiva (izquierda), superior (centro) e interior de auto (derecha)

Luces

Por último se tiene un menú de luces ambientales y direccionales que permiten modificar si están encendidas o no, color e intensidad.

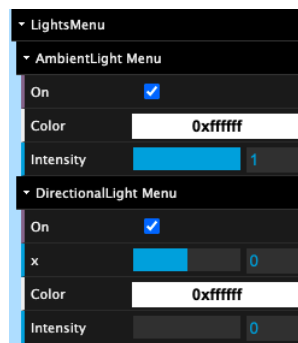


Imagen 1 Menú de luces ambiental y direccional