



Reflexión Individual

Jonatan Hernández García - A01653004

Modelación de sistemas multiagentes con gráficas computacionales

Profesores:

Dr. Sergio Ruiz Loza

Dr. David Christopher Balderas Silva

Fecha de entrega: 1 de diciembre del 2021

Ya que para la resolución del reto se requiere de buscar la ruta más corta de un punto a otro, el modelado requiere de un acercamiento complejo si se busca hacerlo de forma dependiente entre todas las variables es por ello que se eligió un modelo que contiene un entorno el cual se encarga de comunicar a todos los agentes que solicitan información para funcionar. Este considera 2 agentes (Coche e Intersección).

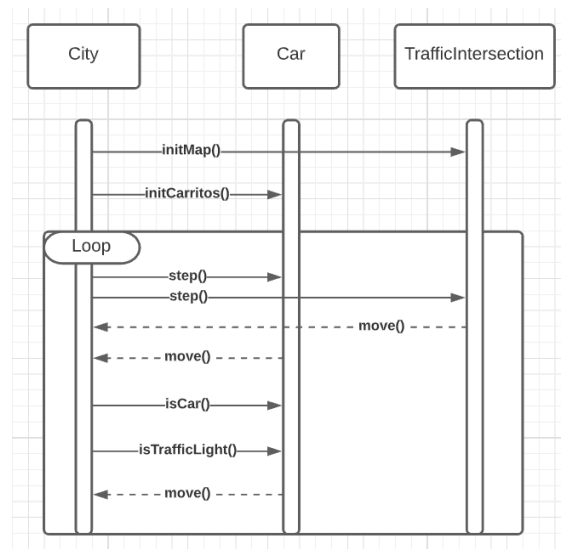


Diagrama de procesos

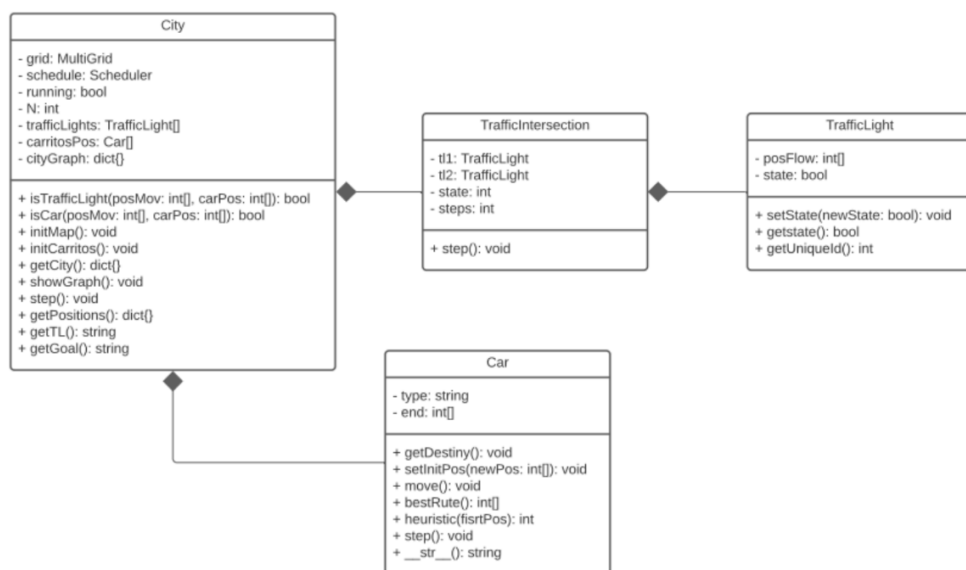
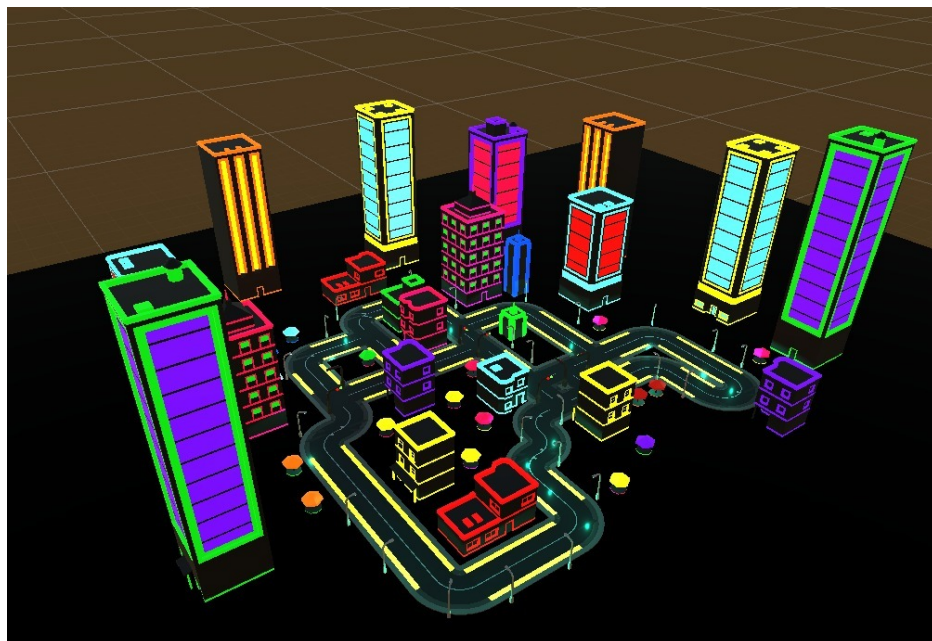


Diagrama de clases

Las variables principales que se tomaron en cuenta para tomar la decisión fueron la posición, dirección, velocidad, destino y rotación de los coches (agentes), la posición y estado de los semáforos, y la distancia entre los diferentes nodos. Todas las antes mencionadas son necesarias para que el modelo funcione correctamente. Para que los coches pudieran rotar, el algoritmo necesita de su posición y dirección. Para que el coche tuviera su dirección, se necesita de su rotación y velocidad. Para que el agente decida si avanzar o no requiere de la posición de otros agentes, del estado del semáforo y la posición de las intersecciones.

El diseño gráfico mostrado se eligió nocturno para lograr un ambiente inmersivo y la atención se centrara en los pocos pero luminosos objetos que se tienen. En caso de que algunos objetos distrajeran al espectador del flujo de coches, se tiene la tecla 0 que quita todos los objetos que no afectan al modelo de multiagentes. Finalmente para que el circuito se pudiera apreciar de diferentes perspectivas se tienen 7 cámaras que se cambian con las teclas 1-7. Desafortunadamente los conceptos que aprendimos para el texturizado y UVs no encajaban tan bien con el diseño ya que apenas comprendimos los conceptos básicos y únicamente aplicamos estos conceptos a las calles del circuito.



Vista en editor de la ciudad

Las ventajas que puedo concluir con nuestra solución es que es eficiente y funcional ya que se usa el algoritmo A* con Dijkstra solo cuando el agente debe tomar una decisión (elección de calles) y hace el recorrido únicamente de las posibles opciones hasta la siguiente intersección. De esta forma podemos decir que el recorrido del algoritmo es realmente necesario cada vez que se aplica.

Algunas desventajas de nuestro modelo es que no está tomando en cuenta calles bidireccionales, tampoco una gran variedad de intersecciones automovilísticas,

también que no es aplicable en zonas no pavimentadas y en ocasiones entra en un loop si no se descongestiona un camino. Finalmente en caso de que las calles de una ciudad estén mal diseñadas y para llegar a un camino la elección es muy lejana una de otra se puede llegar a tener un tiempo indefinido de respuesta.

Para la solucionar la desventaja de bidireccionalidad se retomaría la solución que se tuvo anteriormente, la cual consistió en grafos con direcciones específicas, sin embargo, esta vez la se acoplaría para que funcione para el cálculo del peso de las calles y peso de los coches que anteriormente nos era imposible saber. A simple vista no considero que sea una tarea difícil diferentes intersecciones adicionales, solo tendríamos que acoplar los diferentes casos para cada uno de ellas. Por último para atacar el problema del tiempo indefinido se puede guardar el cálculo de la distancia en el nodo y únicamente pedir el peso vehicular en el camino.

Al inicio del proyecto tuve dificultades para entender cómo cada agente desarrollaba su comportamiento tomando los datos del entorno. Anteriormente había experimentado con la herramienta deep learning que Unity dispone (ml-agents) así que tenía una idea de cómo podría funcionar. Sin embargo, me di cuenta de que mi enfoque era equivocado ya que no se buscaba un deep learning, sino, un machine learning que fuera capaz de que todos sus agentes funcionaran eficientemente en el ambiente de desarrollo. Aunque me hubiera gustado mucho más aprender directamente deep learning, creo que machine learning era una introducción necesaria del tema para que pueda entender a profundidad la inteligencia artificial.

Fue un reto que en definitiva me enseñó el cómo no hacer las cosas y que me hubiera gustado disponer de mucho más tiempo para poder desarrollar una solución más compleja y que me satisficiera más.