Mateo Gonzalez Cosio Ríos y Valles
A01023938

Análisis y Reflexión

Análisis

El reto propuesto para la matera de Modelación de Sistemas Multiagentes con Graficas Computacionales fue construir un modelo multiagente que representara la movilidad urbana en la cuidad de México. Este se refiere, a que debemos crear la representación de una cuidad en donde varios automóviles se dirijan a cierto destino y dentro de su camino se encuentren con intersecciones y varios otros automóviles.

Para solucionar este reto, las herramientas que se proporcionaron en la materia fueron el módulo de Python Mesa, que nos ayuda a construir modelos multiagentes y el programa Unity que nos ayuda a visualizar el modelo. Con estas herramientas definidas empezamos la construcción de nuestro modelo tomando como base la tarea integradora que previamente realizamos. Esta tarea en la que nos basamos consistía en hacer un modelo en donde un grupo de agentes, que representan a varios robots roomba, tenían como objetivo encontrar y limpiar celdas que estaban marcadas como sucias. Esta base fue de gran ayuda para nosotros ya que ciertas funciones que realizamos para esta tarea fueron reutilizadas para conformar la solución del Reto.

Para el modelo establecimos 5 diferentes tipos de agentes, los cuales, cuatro de ellos nos ayudaran a representar la cuidad y el faltante representara los automóviles presentes dentro de la simulación. Los agentes que tenemos para representar la ciudad son: “TrafficLightAgent” (Semáforo), “Road” (Calle), “Obstacle” (Edificio) y “Destination” (Destino de los automóviles). Estos agentes realmente no sufren de muchos cambios dentro de la simulación, y existen simplemente para representar el ambiente en donde nuestro agente principal, siendo el automóvil, estará presente.

Moviéndonos a nuestro agente principal “CarAgent” (Automóvil) el cual fue construido un poco en base a el agente de “Roomba” de la tarea pasada. Este agente fue construido de tal manera que se inicialice con un destino fijo y utilizando una versión del algoritmo A*, encuentre un camino para llegar a dicho destino. Con este camino establecido, el agente cada paso del modelo podrá evaluar el ambiente que lo rodea y comportarse como un automóvil real, haciendo paradas si existe un semáforo que indique alto o si un coche se encuentra parado enfrente de él. Con esto, también

implementamos la funcionalidad de que el automóvil se cambe de carril dentro de una calle con carriles múltiples para poder facilitar una vuelta.

Ya establecidos estos agentes, se creo el modelo para poder llevar acabo esta simulación. Dentro de este modelo, para poder crear la cuidad se lee un archivo de texto que contiene la representación de la cuidad indicando con un carácter diferente en donde deben estar colocados todos los agentes que construyen el ambiente (Calle, Edifico, Simulación y Destino). Dentro de este archivo también se especifica la circulación de la ciudad. Ya leído el archivo, el modelo crea el ambiente dentro de una cuadricula y coloca los agentes dentro.

Aparte de esto el modelo contiene la definición de ciertas funciones que nos ayudan a identificar los vecinos que los agentes Road, la cantidad de movimientos que hizo cada agente, el tiempo necesario para que todos los automóviles lleguen a su destino, obtener las posiciones y atributos de los agentes CarAgent y TrafficLightAgent y para parar la simulación.

En general así esta compuesto nuestro modelo, y aunque dentro del mismo modulo que utilizamos para construirlo podríamos crear una visualización, se nos pidió que utilizáramos la herramienta de Unity para cumplir esto. Para lograr esto, creamos un servidor utilizando el módulo de Python Flask con el objetivo de que funcionara como puente entre Unity y nuestra simulación. Dentro de este servidor, establecimos ciertas rutas que nos permiten configurar el modelo con la cantidad de agentes que queramos, controlar el paso en el que el modelo avanza y obtener información sobre los agentes CarAgent y TrafficLigths.

Con estos enlaces ya definidos creamos un script en Unity el cual se conecta a estos diferentes enlaces para poder crear y mover a los agentes. Pero como dije anteriormente, solo obtenemos la información de los agentes CarAgent y TrafficLights. Esto es, ya que los demás agentes no sufren de cambios visuales se crean utilizando un script que se nos proporciono por los profesores. Este script sigue la misma lógica que el modelo utiliza para crear el ambiente, en donde nuestros agentes principales actúan. Se lee un archivo de texto y se crea la visualización representando cada carácter como un objeto.

Al terminar esto, podríamos desplegar nuestro servidor de manera local y la solución estaría completada, pero se nos pidió que utilizáramos la plataforma de IBM Cloud para alojar nuestro servidor. Para lograr esto tuvimos que crear una cuenta de IBM y descargar la herramienta de

Cloud Foundry CLI, que nos permite desplegar nuestro servidor a dicha plataforma. Antes de desplegar nuestro servidor a la plataforma, tuvimos que hacer una pequeña configuración y agregar algunos archivos a nuestro proyecto, pero ya con esto completado, nuestro servidor se desplego correctamente.

Con esto nuestra solución estaría completa, y podemos identificar con claridad las ventajas y desventajas que se presentan al evaluar nuestra solución. Empezando con las desventajas, podríamos decir que el desplegar nuestro servidor en la nube no es lo mas eficiente, ya que la demora que se presenta entre que el script de Unity pide la información y el momento en la que la recibe es muy alta, la animación de nuestros agentes se ve un poco cortada. Esta desventaja se puede mejorar desplegando el servidor de manera local, ya que la demora disminuiría, y si es que nos quisiéramos deshacer de la demora por completo se podría utilizar el servidor del modulo Mesa para poder hacer la visualización. La otra gran desventaja que existe dentro de nuestro modelo es la inconsistencia de comportamiento de algunos agentes. A través del desarrollo nos hemos dado cuenta de que existen momentos en donde los agentes deberían actuar de manera paralela, pero se puede notar como uno empieza actuar primero que otro. Al darnos cuenta de esto, decidimos utilizar la función de “SimultaneousActivation” del modulo Mesa, que en teoría debería solucionar este problema, pero no es de todo el caso. Una posible solución sería reducir el número de agentes, ya que nos hemos dado cuenta de que este problema se hace notar cada vez más si subimos el número de agentes presentes en la simulación. Fuera de esto no estoy muy seguro como podríamos solucionar esta desventaja.

Siguiendo con las ventajas mas grandes, podemos decir que una de estas es, ya que el modelo este hecho de tal manera que construye el ambiente por medio de un archivo de texto, una gran ventaja es que el ambiente se puede modificar sin tener que tocar ni una línea de código. Con que creemos el nuevo ambiente utilizando los mismos símbolos, el modelo debería interpretarlo y crear la simulación. Otra gran ventaja que existe dentro de nuestra solución es que, ya que elegimos hacer la visualización dentro de Unity, podemos crear una visualización que representa de mejor manera la situación que estamos buscando mostrar. Si hubiéramos decidido hacer la visualización con el modulo Mesa, todos nuestros agentes tendrían la forma de círculos o cuadrados. Dentro de Unity si podemos representar de manera correcta el automóvil, el semáforo, los edificios y las calles.

Mateo Gonzalez Cosio Ríos y Valles
A01023938

Reflexión

En mi opinión, este bloque ha estado excelente y he aprendido mucho mas de lo que me esperaba. Me pareció que este bloque fue muy buena introducción a la concentración de Inteligencia Artificial y un poco a el área de Desarrollo de Videojuegos. Lo único que podría decir que no me gusto tanto fue la duración. Teniendo en cuenta el bloque pasado, me hubiera gustado que la duración de este bloque hubiera sido mas larga y que la duración del bloque pasado mas corta. Esto lo digo ya que gran parte de lo que me hizo disfrutar este bloque fueron las sesiones de teoría, y estas solo duraron dos semanas dando las semanas restantes enfocarnos en la solución de la tarea integradora y el reto. Hubiera disfrutado adentrarme un poco mas en los temas, especialmente en los que estaban relacionados con graficas. Fuera de esto, me gustó mucho la transición entre la tarea integradora y el reto, ya que siento que mucho de lo que aprendimos e hicimos dentro de esta tarea, nos ayudo a desarrollar la solución para el reto. Mirando las expectativas que describí al iniciar el bloque siento que en su mayoría logramos cumplir todas estas y me quedo ansioso por continuar estudiando esto temas dentro de los próximos semestres.