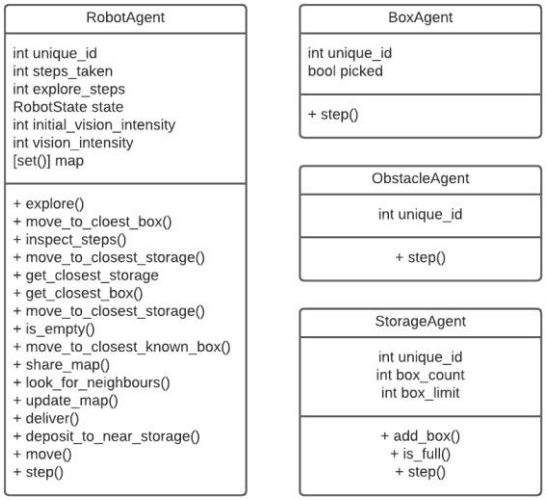
Evidencia 1. Actividad Integradora

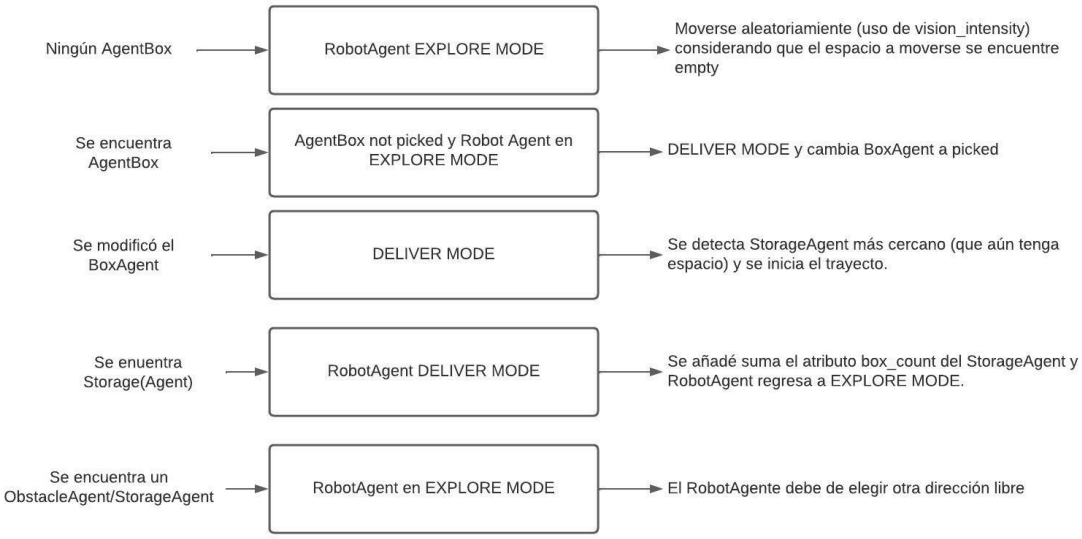
Stephan Guingor Falcon A01029421
Salvador Salgado Normandia A01422874

Para esta Evidencia se nos pidió diseñar un modelo el cual simulará un centro de depósitos, en donde robots tienen la tarea de recolectar cajas para llevarlas a los diferentes puntos apliarlas. De igual manera se debe de considerar que se encuentran en una zona restringida, por lo que los robots están limitado a un espacio específico. Antes de poder dar pie a la simulación de dicha situación es necesario identificar los agentes que van a ser requeridos (protocolos de agentes), diagramas de clases al igual que definir la estrategia en que los agentes podrán cooperar para facilitar las tareas.

Diagramas de clases



Protocolo de agentes



Estrategia de cooperación

En este caso contamos con 4 agentes que cumplen diversas funciones. Para el caso de RobotAgent, este cuenta con la tarea de explorar el mapa en busca de BoxAgents y regresarlos a los diferentes StorageAgents si es que aún tienen espacio de almacenamiento. Este agente realiza funciones según el modo en que se encuentre. En EXPLORE MODE, el agente se moverá por el mapa hasta que se encuentre ya sea al lado de un BoxAgent. En un principio consideramos que el movimiento completamente aleatorio podría hacer que el proceso de recolección fuese muy tardado, por lo que se decidió implementar el atributo de **vision_intensity**. Este último permite que el RobotAgent pueda ver n espacios desde su posición actual (definido por vision_intensity). Esto el permitirá obtener un score por cada posible paso a realizar para hacer que la recolección de caja se más *inteligente*. Al momento de detectar un BoxAgent, el RobotAgent le modificara su atributo de picked a True, haciendo que ya no sea visible para el resto de los agentes. Esto se podría decir que borra al agente del mapa sin removerlo completamente. Una vez realizado esto el RobotAgent cambiará a DELIVER MODE, en donde se definirá la ruta más rápida hacia el StorageAgent más cercano y que cuente con espació suficiente para recibir cajas. Una vez que llegué al StorageAgent, el RobotAgent modificará el atributo de box_count de dicho agente para simular que la caja fue entregada. Acto seguido el RobotAgent cambiará nuevamente a EXPLORE MODE y continuar con la búsqueda.

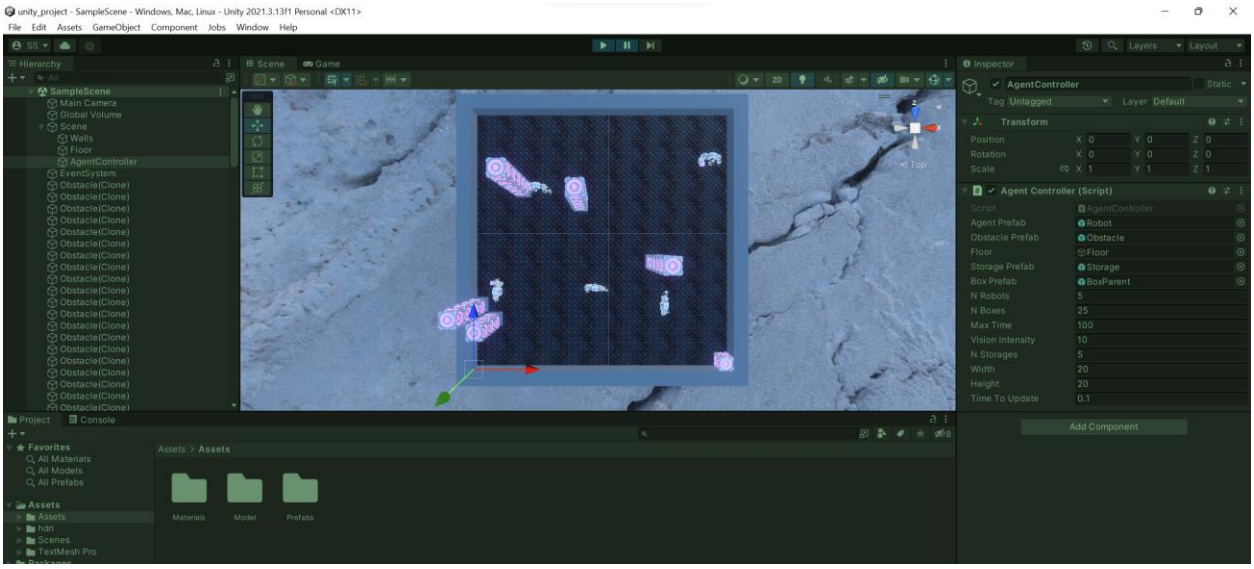
Es importante destacar que durante las pruebas del modelo pudimos percatarnos que a mayor vision_intensity los RobotAgent lograban recolectar las cajas ubicadas cerca del centro del mapa de manera más rápida, pero al sólo quedar caja en las esquinas los agentes permanecían en el centro del mapa sin ir a explorar a los bordes. Principalmente esto se debía que, al tener una gran visión del mapa, esto veían hasta los AgentsObstacles y optaban por el step que les diera un mejor score (se define para cada posible step según varios calculos). Al percatamos de esto se nos ocurrió la idea de que los RobotAgents iniciaran con un **vision_intensity** específico, pero al llegar a n explore_steps sin encontrar ninguna caja su visión disminuye en 1. De esta forma podemos asegurarnos que pueda encontrar todas las cajas aun cuando no sea de la manera más efectiva. Al momento de encontrar una caja (DELIVER MODE) y llevarla al StorageAgent (regreso a EXPLORE MODE) su visión regresaría a la inicial.

Hacía un agente más inteligente

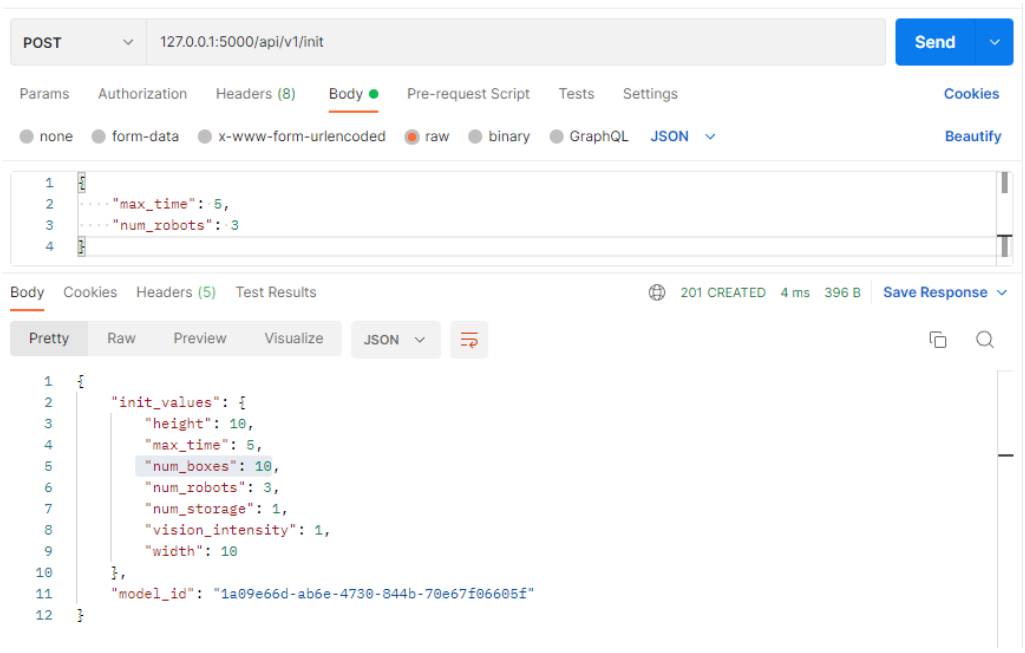
Durante la explicación de esta activad el profesor Octavio nos invitó a ir más allá de las especificaciones de la actividad, buscando que los agentes obtuvieran un comportamiento lo más inteligente posible, por lo que nos dimos a la tarea de pensar en maneras de implementar nuevas funcionalidades al sistema. Pensando que cada agente representa a un robot, se nos hizo sensato pensar en la posibilidad de que parecido al bluetooth, nuestros agentes tuvieran la oportunidad de compartir información los unos con los otros al entrar en contacto. Con esto en mente añadimos un nuevo atributo llamado **known_boxes**, el cual es un diccionario en donde se van registrando las posiciones de cajas validas.

Al momento en que un RobotAgent se encuentra al lado de otro, estos comparten actualizan los BoxAgents que conocen. Gracias al uso de sets eliminamos la posibilidad de repeticiones. Al contar con esta tabla los agentes pueden ir actualizando sus datos tanto en DELIVER MODE como EXPLORE MODE.

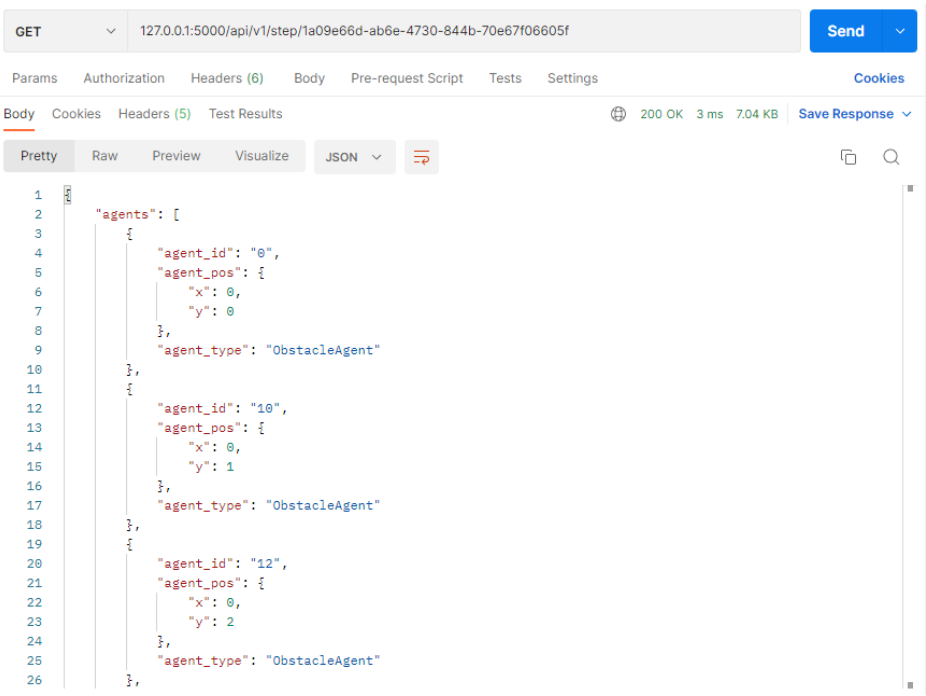
Una de las ventajas de contar con este mapa es que el EXPLORE MODE puede tener dos formas de movimiento. La primera de ellas es la forma ya mencionada anteriormente, pero en caso de contar con posiciones de RobotAgents válidas este se mueve directamente a la coordenada (similar al cómo se mueve en DELIVER MODE). Esta funcionalidad se ve mejorada ya que incluso en DELIVER MODE, el RobotAgent puede identificar BoxAgents para regresar por el más adelante.



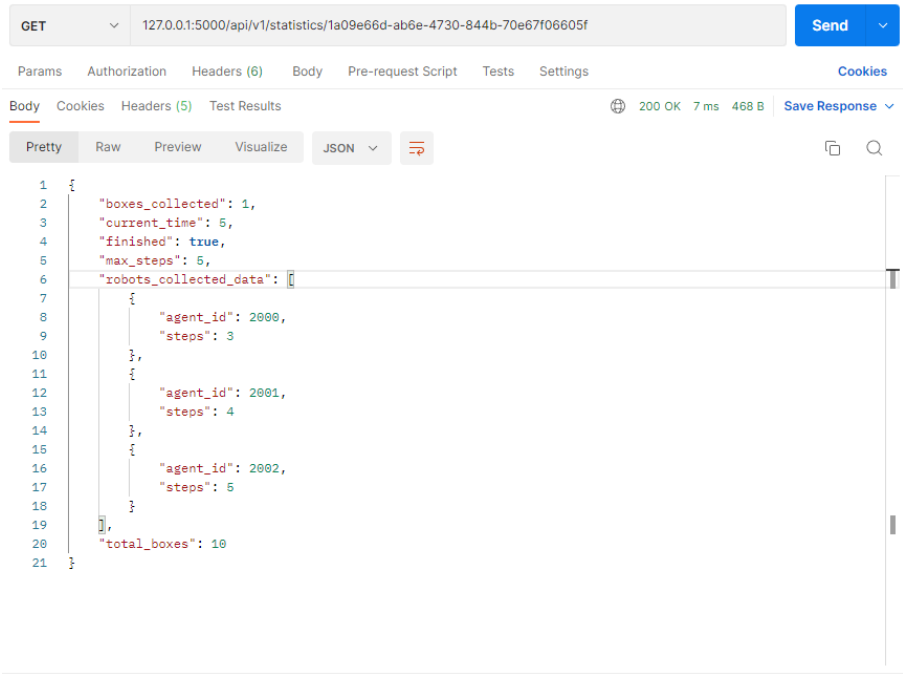
Par los temas de estadísticas diseñamos un endpoint el cual regresa un JSON en donde contine los resultados de la simulación para cada RobotAgent (número de cajas recogidas, número de pasos realizados, etc.)



Para cada modelo nuevo se genera un model_id



Se usa el model_id para obtener la información de los agentes por step



Se obtiene las estadísticas en general