

Sebastián González, A01029746

Ana Paula Katsuda, A01025303

Karla Mondragón, A01025108

Andreína Sanáñez, a01024927

14 de Noviembre del 2022

## **Reporte Actividad Integradora**

### **Descripción del problema.**

5 robots y un almacén lleno de cajas, cada robot está equipado con ruedas omnidireccionales y, por lo tanto, puede conducir en solo cuatro direcciones. Pueden recoger cajas en celdas de cuadrícula adyacentes con sus manipuladores, luego llevarlas a otra ubicación y construir pilas de hasta cinco cajas. A los robots se les permite recibir datos de sensores de las cuatro celdas adyacentes por lo que es fácil distinguir si un campo está libre, es una pared, contiene una pila de cajas (y cuantas cajas hay en la pila) o está ocupado por otro robot. También tienen les indican si llevan una caja en ese momento.

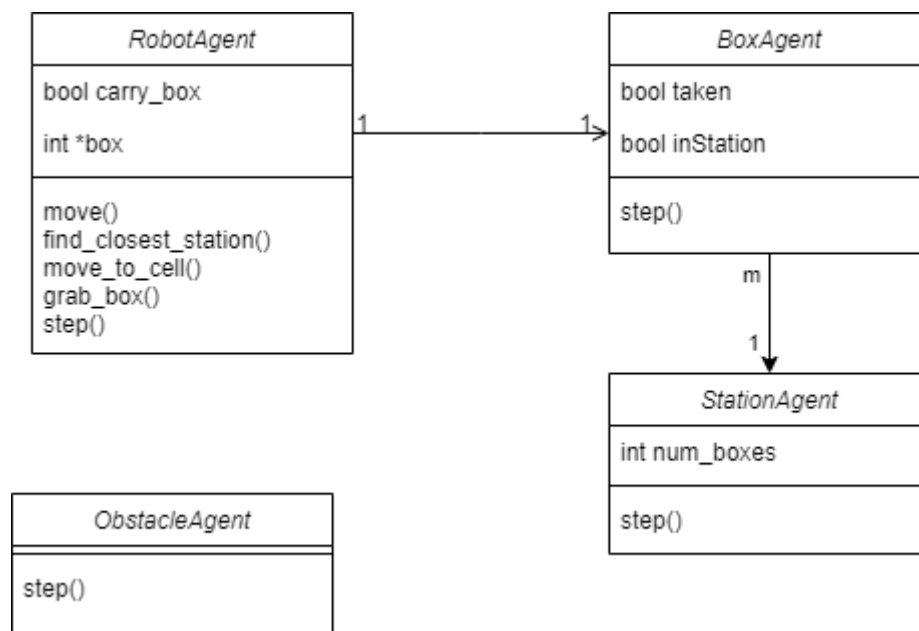
### **Reglas de la simulación:**

- Inicializa las posiciones iniciales de las K cajas. Todas las cajas están a nivel de piso, es decir, no hay pilas de cajas.
- Todos los agentes empiezan en posiciones aleatorias vacías.
- Se ejecuta el tiempo máximo establecido.
- Los agentes buscan cajas, y las llevan a lugares establecidos para apilarlas.
- Se deberá recopilar la siguiente información durante la ejecución:
  - Tiempo necesario hasta que todas las cajas estén en pilas de máximo 5 cajas.
  - Número de movimientos realizados por todos los robots.
  - Estrategia que podría disminuir el tiempo dedicado, así como la cantidad de movimientos realizados.

### **Diagrama de Clases**

Dentro de la simulación existen cuatro agentes en total: los robots, las cajas, las estaciones y los obstáculos (el borde). De dichos agentes, solamente uno es activo y se mueve, los robots, todos los demás son estáticos y las cajas solamente pueden cambiar de posición si el robot las mueve.

Con respecto a los atributos de cada agente, en el caso de *RobotAgent*, solamente tiene dos atributos que consiste en una variable booleana `carry_box`, que describe si un robot está moviendo una caja o está libre y una variable que apunta a la caja que está cargando. *BoxAgent* tiene dos variables booleanas, la primera se refiere a si la caja en sí ya ha sido tomada y la segunda que describe si la caja ya se encuentra en una estación. Por último, *StationAgent* solamente tiene una variable que contiene el número de cajas que se encuentran en la estación.



Todos los agentes, por excepción a *RobotAgent*, solamente tienen el método `step()` en donde se indica que estos no se mueven por su cuenta a través de la simulación. *RobotAgent* tiene el método `move()` que dependiendo de las condiciones a su alrededor le permite moverse solamente en cuatro direcciones, `find_closest_station()` que hace los cálculos para encontrar la estación disponible más cercana, `move_to_cell()` que recibe las coordenadas de la estación disponible más cercana y a partir de eso elige la celda más cercana a la estación, `grab_box()` que agarra la caja y `step()` que llama a `move()`.

### Resultados de la Simulación

Tanto el tiempo necesario para que todas las cajas estén apiladas como el número de movimientos realizados por todos los robots, dependen del número de cajas que se inicializan en cada prueba de la simulación. Ambos resultados, se exponen a través de la consola de Unity.

Con respecto a la estrategia o práctica para disminuir el tiempo dedicado o la cantidad de movimientos realizados, consideramos que se podría implementar una estrategia de cooperación entre los robots, en otras palabras que se puedan comunicar entre ellos para que puedan compartir información. Un ejemplo de lo anterior sería que cada robot evaluará la cantidad de cajas que se encuentra alrededor del mismo (en sus celdas vecinas), donde si esta es mayor a dos, le avisa a los robots que se encuentren dentro de un radio de cercanía y se dirija el más cercano a las coordenadas donde hay cajas que recoger. Otra estrategia puede ser que conforme el robot vaya avanzando hacia la estación y encuentre cajas en su camino, guarde la posición de las otras cajas para que una vez que llegue a la estación vaya directo a las cajas que ya ha encontrado en vez de constantemente checar en las celdas vecinas por cajas, esta estrategia puede ser complementada con el primer ejemplo.