



Instituto Tecnológico de Estudios Superiores de Monterrey

Modelación de sistemas multiagentes con gráficas computacionales

Grupo 101

Equipo Druids

Arturo Ramos Martínez A01643269

Adolfo Hernández Signoret A01637184

Bryan Ithan Landín Lara A01636271

Diego Enrique Vargas Ramírez A01635782

Luis Fernando Cuevas Arroyo A01637254

Campus Guadalajara

Lunes 26 de agosto de 2024

- **Especificaciones de propiedades de agentes (python - atributos)**
 - **RobotAgent:**
 - **AgentType:** Tipo de agente, en este caso, se establece en 0.
 - **Pos:** Posición actual del robot en la cuadrícula.
 - **Target:** Posición objetivo que el robot planea alcanzar.
 - **Has_box:** booleano que indica si el robot está cargando una caja.
 - **Reserved_position:** Posición reservada por el robot en la cuadrícula, utilizada para evitar colisiones.
 - **First_step:** booleano que indica si el robot está en su primer paso de ejecución.
 - **Original_pos:** Posición original del robot al inicio de la simulación.
 - **Movement_history:** Contador de los pasos del robot.
 - **BoxAgent:**
 - **AgentType:** Tipo de agente, en este caso, se establece en 1.
 - **Has_been_picked:** Booleano que indica si la caja ha sido recogida por un robot.
 - **Pos:** Posición actual de la caja en la cuadrícula.
 - **ContainerAgent:**
 - **AgentType:** Tipo de agente, en este caso, se establece en 2.
 - **Capacity:** Entero que indica la capacidad inicial del contenedor.
 - **Pos:** Posición del contenedor en la cuadrícula.

- **Especificaciones de acciones y reglas de agentes (python - métodos)**
 - **RobotAgent:**
 - **Setup:** Se inicializan los procesos del agente.
 - **See:** Se analizan los elementos en el ambiente.
 - **Next:** Se avanza a la siguiente acción del agente.
 - **Step:** Pasos que sigue el agente.
 - **Update:** Se actualiza la información del agente.
 - **End:** Se termina la ejecución de acciones del agente.
 - **pick_target:** tomando en cuenta las distancias y haciendo uso de la fórmula heurística, obtenemos la caja óptima a recoger.
 - **reserve_to_target:** Haciendo uso del algoritmo de estrella y el conocimiento de los espacios vecinos, se calcula el siguiente paso que se debe de tomar para tener un recorrido óptimo.
 - **pick_up_box:** elimina el agente de la caja, modifica sus creencias para indicar que tiene una caja, y elimina el target que tenía.
 - **pick_target_container:** escoge un contenedor hacia el cual moverse.
 - **drop_box:** modifica a falso su creencia de que tiene una caja, elimina su target, y la creencia del agente contenedor en el cual se libera la caja, cambia su capacidad por -1.
 - **return_to_base:** indica que el target a seguir es la posición en donde inició originalmente.
 - **pick_target_rule:** se valida si no hay un target, si no tiene una caja, y si aún quedan cajas por recoger.
 - **reserve_to_target_rule:** se valida si se tiene un target y si no se encuentra ahí actualmente.
 - **pick_up_box_rule:** se valida si se recogió una caja.
 - **pick_target_container_rule:** se valida si no tiene un target y si tiene una caja.

- **drop_box_rule:** se valida si tiene un target y si se encuentra en esa posición, así como debe de también tener una caja.
 - **return_to_base_rule:** se valida si ya no hay cajas por recoger y no tiene una caja.
 - **BoxAgent:**
 - **Setup:** Se inicializan los procesos del agente.
 - **See:** Se analizan los elementos en el ambiente.
 - **Next:** Se avanza a la siguiente acción del agente.
 - **Step:** Pasos que sigue el agente.
 - **Update:** Se actualiza la información del agente.
 - **End:** Se termina la ejecución de acciones del agente.
 - **ContainerAgent:**
 - **Setup:** Se inicializan los procesos del agente.
 - **See:** Se analizan los elementos en el ambiente.
 - **Next:** Se avanza a la siguiente acción del agente.
 - **Step:** Pasos que sigue el agente.
 - **Update:** Se actualiza la información del agente.
 - **End:** Se termina la ejecución de acciones del agente.
- **Especificaciones de propiedades de ambiente**
 - **Determinista o No Determinista**
 - **Determinista,** nuestro agente siempre busca la mejor ruta para sus objetivos, así como siempre buscar aquel que sea más cercano. Su comportamiento es deliberado y decidido en base a un análisis de su entorno.
 - **Estatico o Dinamico**
 - El ambiente es **estático** al no estar en cambio fluido sino que por medio de steps, teniendo cada robot un momento dado para deliberar y decidir hacia

dónde dirigirse en base a su condición actual. Sea que esta tenga o no una caja o si nos hemos quedado sin contenedores.

- **Accesible o No Accesible**

- Nuestro ambiente es **accesible**, debido a que el robot en todo momento tiene constancia de las posiciones de las cajas así como de los contenedores. Además de tener código el cual le permite evitar colisiones con los otros agentes robot.

- **Discreto o Continuo**

- El entorno es **discreto** al tratarse de una cuadrícula limitada, la cual tiene lugar dentro de una simulación por pasos y con acciones regidas por reglas las cuales limitan las acciones disponibles en cada momento del agente.

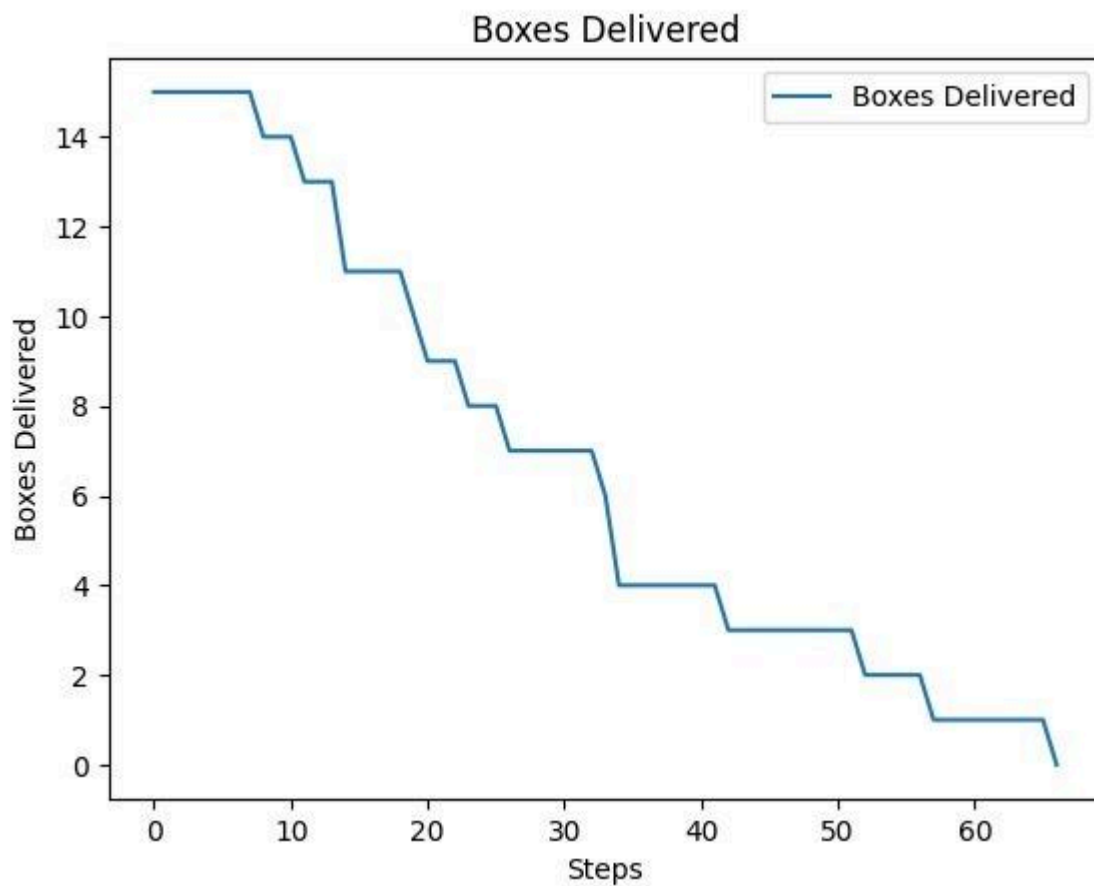
- **Episodico o No Episodico**

- **No episódico**, ya que nuestro agente decide siempre tomar los caminos más sencillos, con el tiempo este se verá sujeto a tomar rutas más largas siendo este hecho una consecuencia de las decisiones pasadas.

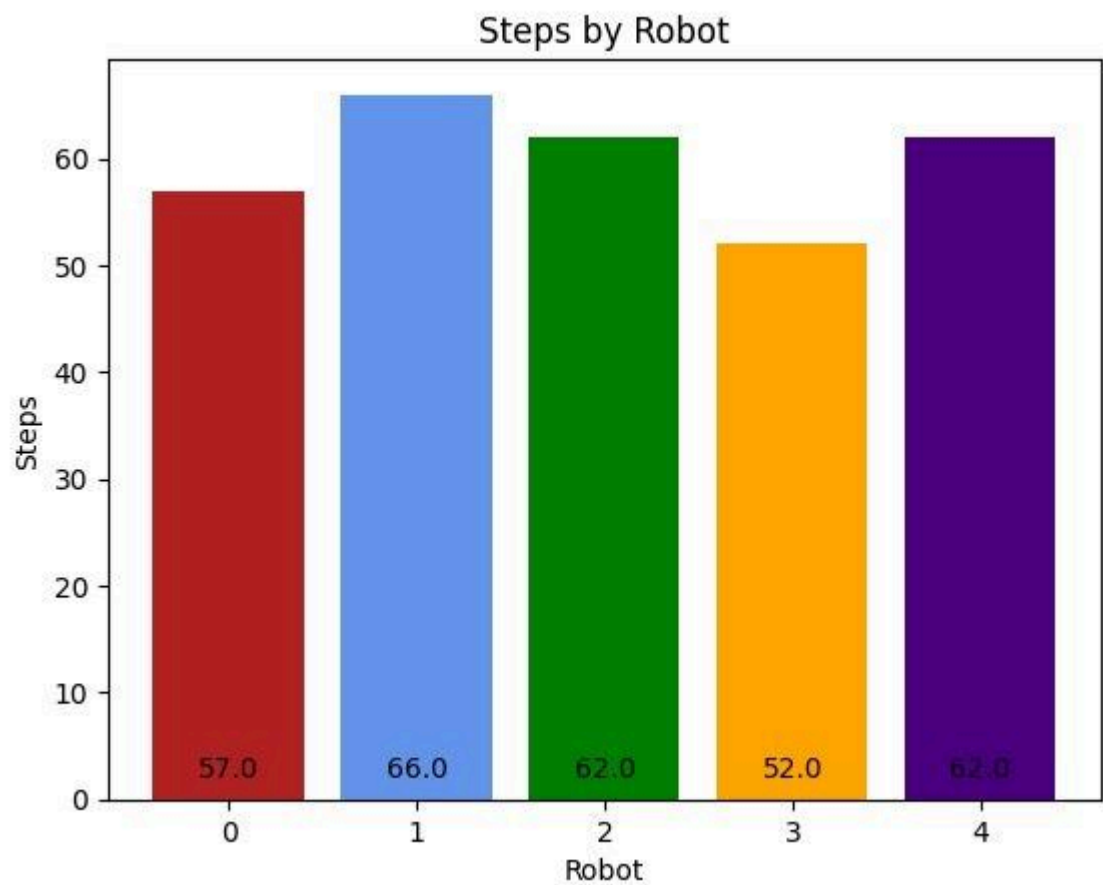
- **Métrica de utilidad o éxito de cada agente**

El éxito para un agente robot consiste en recorrer la menor cantidad posible de pasos antes de que se entregue la última caja, ya que colectivamente, esto indica que fueron capaces entre todos los agentes robots, de hacer un trabajo óptimo.

```
parameters = {  
    'M': 16,  
    'N': 32,  
    'robots': 5,  
    'boxes': 15,  
    'containers': 3,  
    'steps': 100,  
}
```



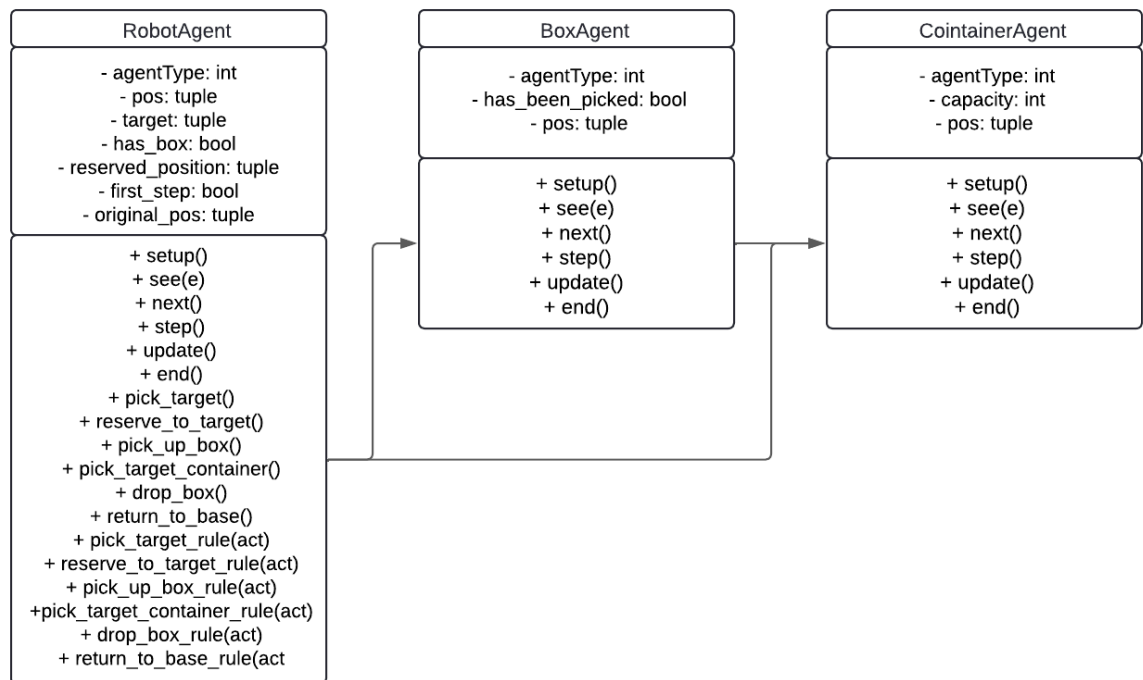
Una métrica para cada agente robot más individualista, puede ser el conteo individual de pasos tomados, si sus pasos son menores a los indicados al comienzo de la simulación, indica que tuvo que lidiar con colisiones en algún momento.



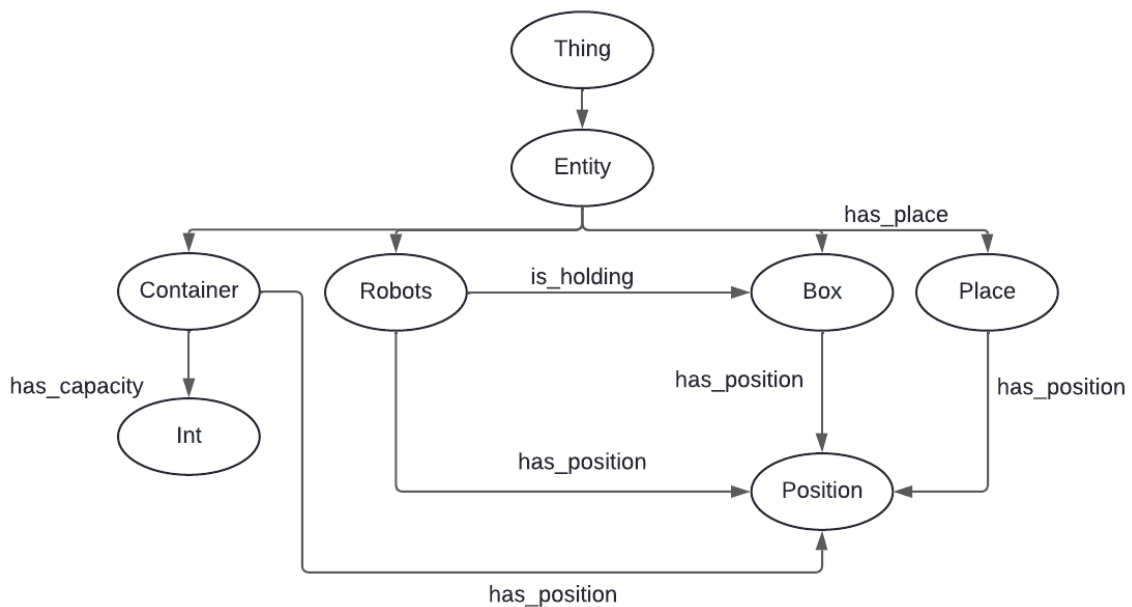
Para los agentes caja, su mayor métrica es el hecho de que su existencia desaparece en la simulación, pues esto indica que ha sido recogida.

Para los agentes contenedores, la comparación entre su capacidad inicial y su capacidad final indica cuántas cajas recibió en la simulación.

- Diagramas de clases de los agentes utilizados



- Diagramas de clase de las ontologías utilizadas



- **Analiza si existe una estrategia que podría disminuir el tiempo dedicado, así como la cantidad de movimientos realizados. ¿Cómo sería? Descríbela**

Primero la estrategia consistió en un algoritmo que ejecuta pasos en direcciones random. Este si bien era un proceso más sencillo para el modelo computacionalmente en el paso a paso, al final terminaba siendo menos eficiente pues requería de muchos más pasos para completar su objetivo pues repetía muchos espacios sin una meta. Después, se implementó un algoritmo de path finding a-star que calculaba la trayectoria completa que debía seguir cada robot hacia su objetivo. Pero esto, debido a que no tomaba en cuenta las rutas de los demás robots, provocaba colisiones. Finalmente se implementó el mismo algoritmo de pathfinding pero ahora solo regresa la siguiente mejor posición, esto permite moverse con flexibilidad evitando colisiones.

Teniendo un algoritmo de backtracking que considere no solo la trayectoria de este agente individual sino de todos se podría obtener las trayectorias definitivas sin colisiones desde el comienzo, de esta manera no solo tendríamos un tiempo menor si no que se harían la menor cantidad de movimientos posibles.

- **Conclusión donde se incluya una breve descripción de soluciones alternativas para mejorar la eficiencia de los agentes.**

Pese a tener excelentes resultados esperados, existen diversas áreas de oportunidad para mejorar el código y la lógica implementada. Una de estas mejoras podría ser implementar un algoritmo de pathfinding más avanzado que considere las trayectorias de todos los robots; por otro lado, sería establecer un sistema de fila para dejar cajas, priorizar clusters de cajas cercanos y asignar zonas específicas de trabajo a cada robot. Estas estrategias que podrían reducir el tiempo de ejecución y el número de movimientos, optimizando así el rendimiento general del sistema. Por otro último, en este entregable, no fue requerimiento la comunicación entre agentes; sin embargo, esto mejoraría en gran medida la eficiencia para la coordinación y reduciría las colisiones.

