



Tecnológico de Monterrey

Evidencia 1. Actividad Integradora

Unidad de formación:

Modelación de sistemas multiagentes con gráficas computacionales
(Gpo 302)

Profesores:

Iván Axel Dounce Nava
Javier Félix Rendón

Alumnos:

A01737220 - Andrea Guadalupe Blanco Rubio
A01637163 - Fernanda Reyes Martínez
A01643383 - Sayid Isaias Valdivia Frausto
A01613911 - Roxana Aranda González Ramírez
A01638587 - Karime Monzerrat Muñoz Franco

Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Guadalajara

Av. Gral Ramón Corona No 2514, Col Nuevo México, Código Postal 45201 Zapopan, Jal.

11 de noviembre de 2024

Propiedades del agente - atributos

- **Agente del robot (RobotAgent)**
 - *agentType*: indica el tipo de agente, tiene un valor entero de 0.
 - *is_carrying*: verifica si el agente carga o no una caja en ese momento, su valor inicial es False y cambia a True cuando recoge una caja.
 - *targetBox*: si observa cajas cercanas o no, se inicializa como None.
 - *path*: movimientos del plan para llegar a la caja, indica cómo moverse en cada dimensión.
- **Agente de la caja (BoxAgent)**
 - *agentType*: indica el tipo de agente, tiene un valor entero de 1.
 - *position*: guarda la posición actual de la caja, se inicializa como None y se actualiza usando un método.
- **Agente de los estantes (ShelfAgent)**
 - *agentType*: indica el tipo de agente, tiene un valor entero de 2.

Acciones y reglas del agente - métodos

- **Agente del robot (RobotAgent)**
 - *setup*: inicializa las propiedades y procesos del agente.
 - *see*: observa el entorno cercano para identificar cajas y estantes.
 - *pick_box*: recoge una caja objetivo si está en la misma posición y no está cargando otra.
 - *move*: avanza siguiendo la ruta planificada.
 - *step*: ejecuta el comportamiento del agente.
 - *plan_path*: calcula la ruta hacia el objetivo y la almacena.
- **Agente de la caja (BoxAgent)**
 - *setup*: inicializa las propiedades y procesos del agente.
 - *set_position*: establece la posición actual de la caja.
 - *step*: pasos que sigue el agente, no realiza acción en esta implementación.
- **Agente de los estantes (ShelfAgent)**
 - *setup*: inicializa las propiedades y procesos del agente.
 - *step*: pasos que sigue el agente, no realiza acción en esta implementación.
 - *update*: actualiza la información del agente, no realiza acción en esta implementación.
 - *end*: termina la ejecución, no realiza acción en esta implementación.

Propiedades del ambiente

- **Determinista o no determinista**
 - El ambiente es determinista porque el comportamiento de los robots sigue reglas, como identificar cajas cercanas y calcular rutas óptimas sin elementos de aleatoriedad.

- **Estático o dinámico**
 - Estático, el estado no cambia de manera continua, se usan steps para que los agentes tomen decisiones y actúen en base a las condiciones actuales.
- **Accesible o no accesible**
 - Los agentes tienen acceso completo a la información de su entorno inmediato, permite planificar sus movimientos, es accesible.
- **Discreto o continuo**
 - El espacio y el tiempo están discretizados en el modelo, los movimientos se limitan y las acciones solo se dan en ciertas iteraciones.
- **Episódico o no episódico**
 - El ambiente es no episódico porque las decisiones de los robots dependen del estado actual del entorno y de las acciones previas.

Métrica de utilidad o éxito de los agentes

Actualmente el código tiene una métrica implementada para robots el número total de movimientos:

```
def end(self):
    totalMovements = sum(robot.movements for robot in self.robots)
    print(f"Movimientos: {totalMovements}")
```

Sin embargo, esta métrica por sí sola sabemos que no es una medida robusta del éxito del robot. Sólo nos está indicando la actividad, no necesariamente eficiencia o logro de objetivos. No hay métricas específicas implementadas para los agentes “BoxAgent” o “ShelfAgent” ya que sus roles en esta simulación son pasivos.

Algunas métricas que pudimos haber implementado en este reto pueden ser las siguientes:

- Cajas entregadas: Contar el número de cajas que cada robot recoge y entrega con éxito en los estantes. Esto se reflejaba directamente en su tarea principal
 - Implementación: Agregar un contador a “RobotAgent” Se incrementa cada vez que la caja se coloca con éxito en un estante.
- Tiempo de entrega: Rastrear el tiempo (número de pasos de simulación) Se tarda un robot en entregar una caja desde la recogida hasta el estante. Esto ayudaría a evaluar la eficiencia.
 - Implementación: Registrar el número de pasos cuando un robot recoge una caja y nuevamente cuando la entrega. La diferencia es el tiempo de entrega.
- Optimización de rutas: Compara la longitud real de la ruta tomada por un robot con la ruta más corta posible. Una proporción más baja indica una mejor planificación de la ruta.
 - Implementación: Se podría necesitar un algoritmo de búsqueda de rutas (como A*) para calcular la ruta más corta. Comparar su longitud con los movimientos reales del robot.

- Colisiones: Contar el número de veces que un robot choca con otros robots u obstáculos. Menos colisiones significan una mejor navegación.
 - Implementación: Agregar un contador a “RobotAgent” Que se incrementa cuando se detecta una colisión en “resolve_collision”

Al implementar estas métricas podríamos haber obtenido una comprensión más profunda de la simulación del almacén y podríamos haber evaluado el rendimiento de los agentes de una manera más efectiva.

Diagrama de clase de la ontología

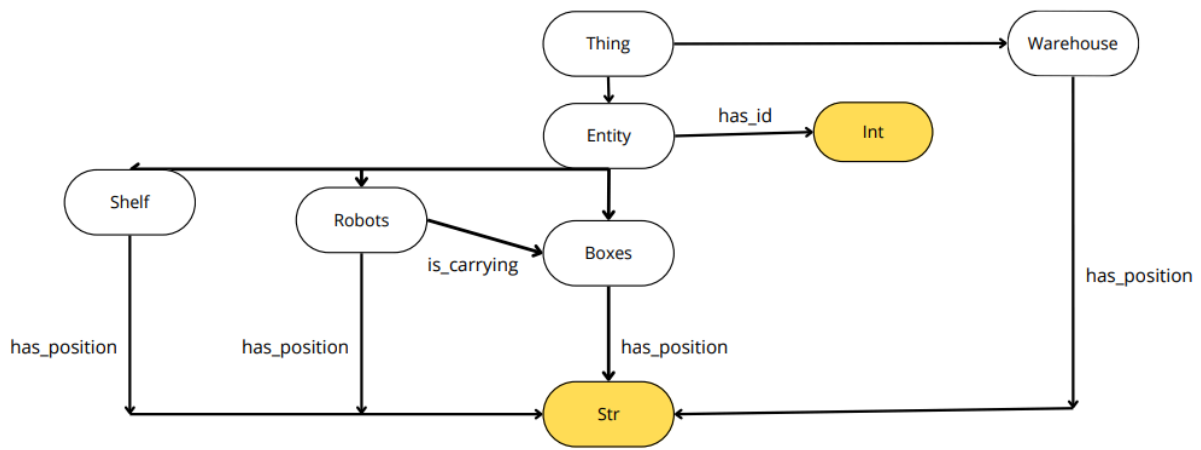
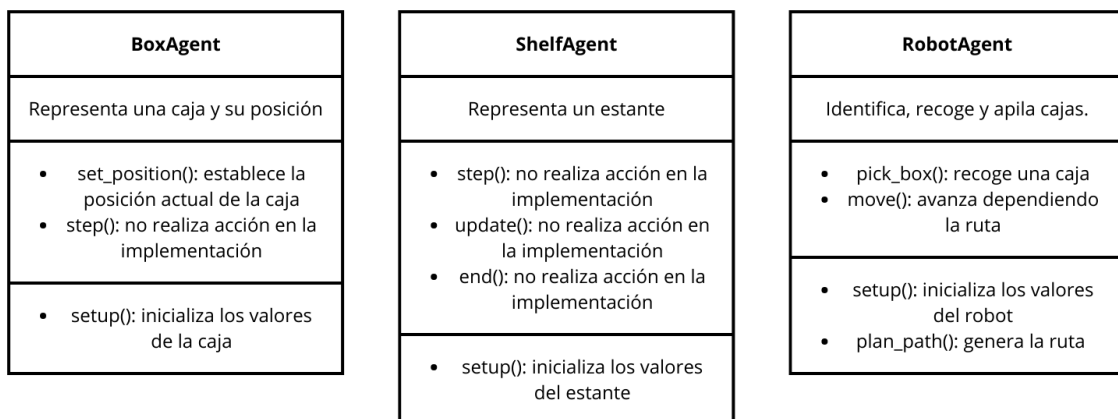


Diagrama de clase de los agentes



Conclusión

Hemos analizado como equipo la necesidad de implementar métricas más robustas para evaluar el éxito y la eficiencia de los agentes. A pesar de presentar una métrica actual, la cual es el número de movimientos totales de los robots, es insuficiente para comprender completamente el rendimiento del sistema.

Sin embargo, en este documento proponemos diversas métricas para cada tipo de agente sobre todo los robots que permitirán un análisis más profundo. Implementar estas métricas, junto con la recolección y visualización de datos, nos proporciona una comprensión más completa del funcionamiento del almacén simulado y nos permitiría identificar áreas de mejora.

Además de las métricas, hemos identificado otras soluciones alternativas para optimizar la eficiencia de los agentes en la simulación:

- Algoritmos de Planificación de Rutas Avanzados: Implementar algoritmos como A* o Dijkstra para generar rutas más óptimas para los robots, minimizando la distancia recorrida y evitando colisiones.
- Asignación de Tareas Dinámica: Asignar tareas a los robots en tiempo real basándose en su ubicación, disponibilidad y la prioridad de las tareas. Esto podría mejorar la eficiencia general del sistema.
- Comunicación entre Agentes: Permitir que los robots se comuniquen entre sí para compartir información sobre la ubicación de las cajas, los estantes y los obstáculos. Esto podría ayudar a evitar colisiones y optimizar las rutas.
- Priorización de Tareas: Implementar un sistema de prioridades para las cajas, de modo que los robots se enfoquen en entregar las cajas más urgentes primero.
- Optimización del Diseño del Almacén: Explorar diferentes configuraciones del almacén, como la ubicación de los estantes y las zonas de paso, para minimizar las distancias de viaje y mejorar el flujo de trabajo.

A pesar de que la simulación del almacén se encuentra en una etapa inconclusa, nuestro equipo ha logrado avances significativos en el desarrollo de un modelo funcional y en la identificación de áreas clave para la mejora. También ha permitido a nuestro equipo adquirir experiencia en la modelación de sistemas complejos, la programación orientada a agentes y la aplicación de conceptos de inteligencia artificial en la resolución de problemas logísticos.

Hemos implementado la estructura básica de la simulación, incluyendo la creación de agentes (robots, cajas y estantes), la definición de sus comportamientos y la interacción entre ellos dentro del entorno del almacén. Además, hemos reconocido la importancia de establecer métricas robustas para evaluar la eficiencia del sistema y la optimización de las rutas de los robots.