



Actividad Integradora

Sergio Manuel Gonzalez Vargas - A01745446

Gilberto André García Gaytán - A01753176

Fernando Ortiz Saldaña - A01376737

Ricardo Ramírez Condado - A01379299

Modelación de sistemas multiagentes con gráficas computacionales

Grupo: 302

21 Noviembre 2022

Instituto de Estudios Superiores del Tecnológico de Monterrey

Profesores:

Jorge Adolfo Ramírez Uresti

Octavio Navarro Hinojosa

Sistemas Multiagentes.

Objetivo: Programar 5 agentes que apilen cajas dentro de un almacén.

Introducción.

Un agente inteligente se refiere a una amplia variedad de software, desde el más simple hasta el más complejo. Puede ser una herramienta de recopilación de información, monitoreo informático o incluso una unidad inteligente; dichas unidades son capaces de realizar actividades muy específicas, sin embargo, como parte de su desarrollo, tienen ciertas limitaciones, pero eso no limita su funcionamiento, por lo que en esta actividad se mostrará un modelado 3d con Unity sobre robots acomodando cajas en un almacén.

Descripción del ambiente

En esta simulación hay representa un modelo 3d donde el usuario es dueño de cinco robots, los cuales son nuestros agentes y que tendrán la tarea de organizar cajas que se encuentran en un almacén, es decir, representar un sistema de multiagentes que organicen cajas.

- Nuestro ambiente es accesible ya que nuestros agentes serán capaces de detectar mediante sus sensores la información que se encuentra en él.
- Nuestro ambiente es determinístico ya que el otro estado de nuestros agentes, está determinado por el estado actual, ya que los agentes conocen algunos estados en los que se encuentran las cajas y esto ayuda a determinar qué es lo que hará cada agente.
- Nuestro ambiente es episódico, ya que las acciones de los agentes están en diferentes tiempos y los episodios siguientes, que no son dependientes a los pasados.
- Nuestro ambiente es estático ya que todo está programado, ya que nosotros definimos lo que harán nuestros agentes, e incluso se indica en dónde deben de acomodarse algunas cajas.
- Nuestro ambiente es discreto, ya que los robots sólo pueden moverse para detectar cajas o acomodarlas.

Descripción PEAS de cada agente

Robot_Agent

Performance: Ser capaz de encontrar por sí mismo en donde se encuentran las cajas y llevarlas a las tarimas.

Environment: Accesible, determinístico, episódico, estático y discreto.

Actuators: El robot detecta cuando ve un caja y la lleva a las tarimas para irlas apilando de 5 en 5.

Sensors: Llevar cajas en el menor tiempo posible a las tarimas, percibiendo cuál es el camino para llegar a las tarimas.

Nota: Es necesario aclarar que en un entorno real, las cajas NO son agentes, sin embargo, para la aplicación y simulación de este prototipo, fue necesario declarar "caja" como

agente, ya que esta arrojará su estatus, es decir, si está acomodada, se está moviendo, si fue recogida, etc. Por ello ayudaría a implementar efectivamente el código.

Cajas

Performance: Ser llevada a las tarimas por el agente robot.

Environment: Accesible, determinístico, episódico, estático y discreto.

Actuators: Toma el estado actual del agente y el siguiente estado del agente y lo mueve del estado actual al siguiente estado.

Sensors: Ser llevado a la tarima mediante estados.

AUML

Robot_Agent
Grupo: Robots
Rol: Robot que moverá cajas
Servicio: Llevar las cajas a las tarimas de 5 en 5
Protocolo: Llevar cajas
Eventos: Buscar el camino para llevar las cajas
Planes: No hay planes
Metas: Llevarla en el menor tiempo posible
Acciones: Llegar de un punto A, a un punto B
Conocimiento: Ser capaz de encontrar por si mismo las cajas y llevarlas hacia las tarimas.

Figura 1. Diagrama de Robot_Agent

Caja
Grupo: Cajas
Rol: Caja que será movida
Servicio: Ser llevadas a las tarimas
Protocolo: Ser llevadas
Eventos: Toma el estado actual del agente y el siguiente estado del agente y lo mueve del estado actual al siguiente estado
Planes: No hay planes
Metas: Ser llevada a destino
Acciones: Llegar de un punto A, a un punto B
Conocimiento: Ser llevada a la tarima tomando el estado del agente robot

Figura 2. Diagrama de Agente Caja

Propuesta de Arquitectura Reactiva

Propuesta	
8	NADA → DAR PASO
7	DETECTAR CAJA EN PERÍMETRO → PREPARAR PARA RECOGER CAJA
6	PREPARAR PARA RECOGER CAJA Y TENGO CAJA → IGNORAR
5	PREPARAR PARA RECOGER CAJA Y NO TENGO CAJA → RECOGER CAJA
4	TENGO CAJA Y NADA → PREPARAR PARA APILAR CAJA
3	PREPARAR PARA APILAR CAJA → BUSCAR ESTANTE
2	APILAR CAJA Y ESTANTE LLENO → BUSCAR OTRO ESTANTE
1	APILAR CAJA Y ESTANTE NO LLENO → COLOCAR CAJA

Figura 3. Arquitectura Reactiva

Diagramas de Protocolo de Interacción

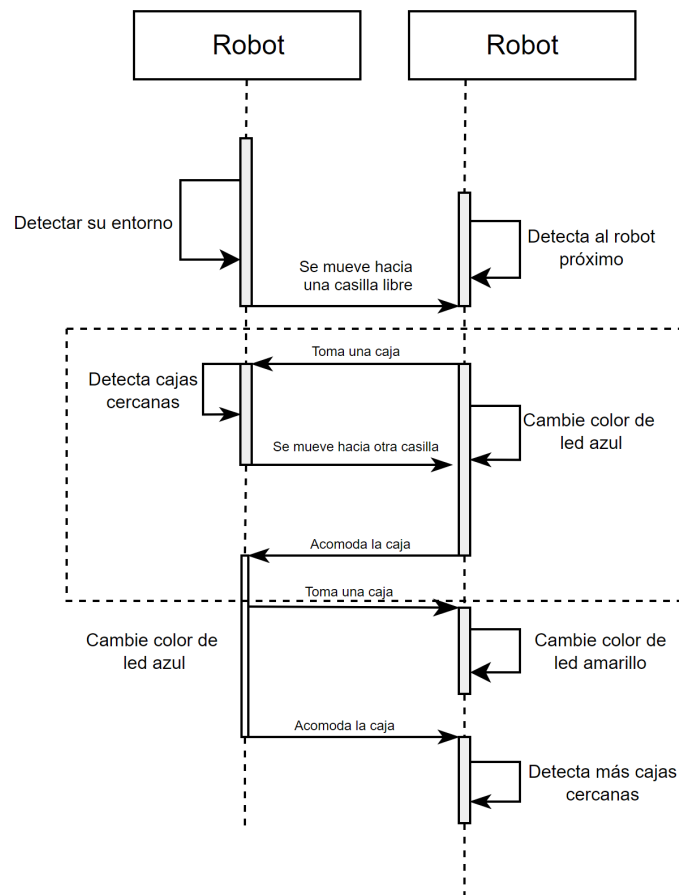


Figura 4. Diagrama de protocolo de interacción Robot-Robot

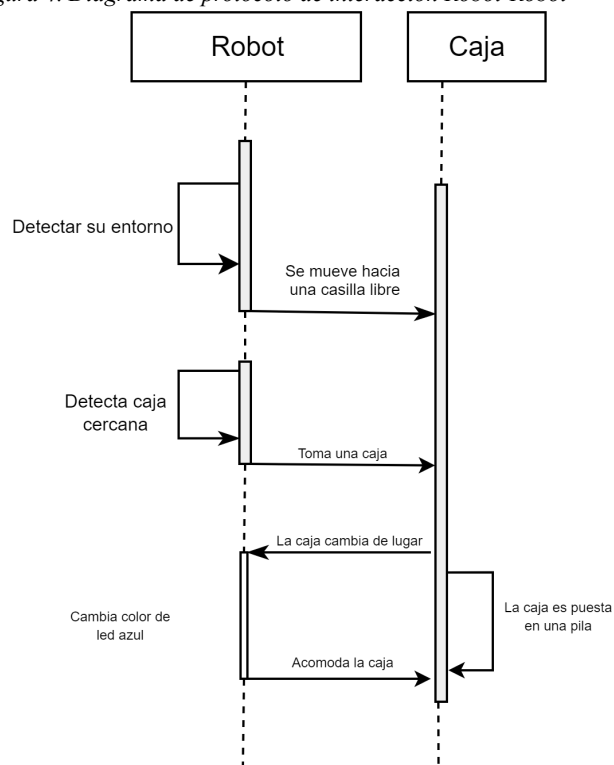


Figura 5. Diagrama de protocolo de interacción Robot-Caja

Diagrama Organizacional

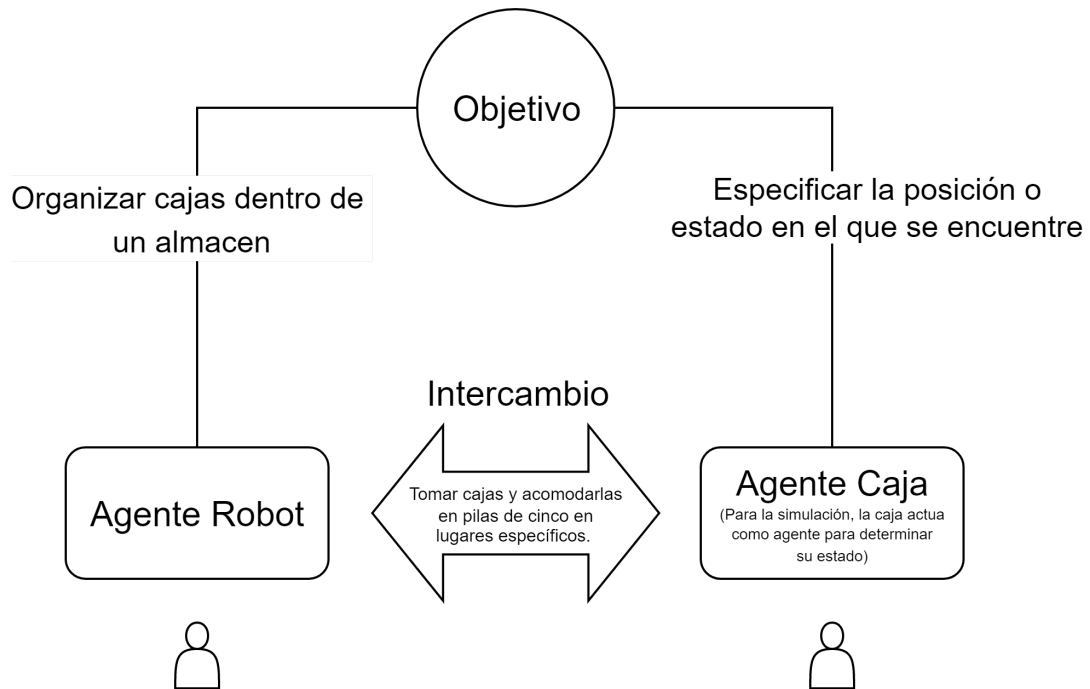


Figura 6. Diagrama organizacional

Analisis de Agentes

- **Tiempo necesario hasta que todas las cajas estén en pilas máximo 5 cajas.**

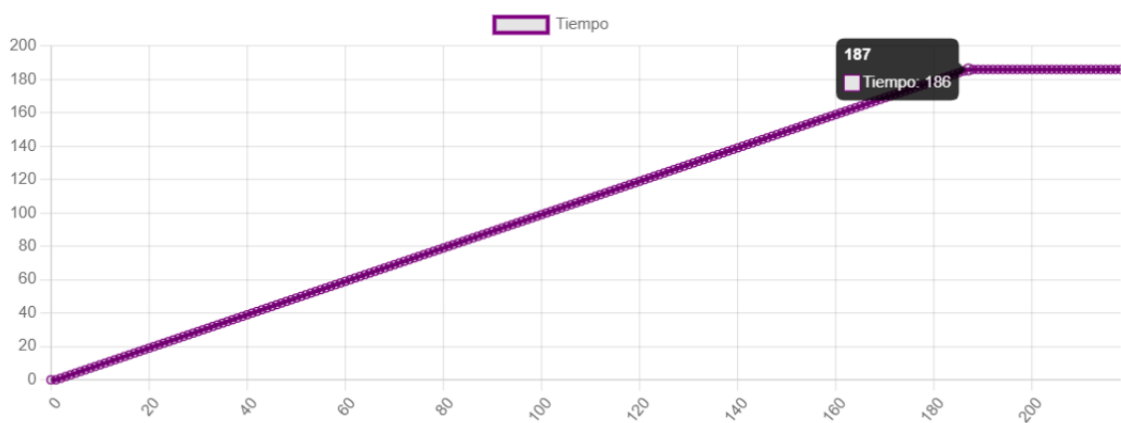


Figura 7. Tiempo tardado en apilar las cajas en mesa (frames de 10 en 10)

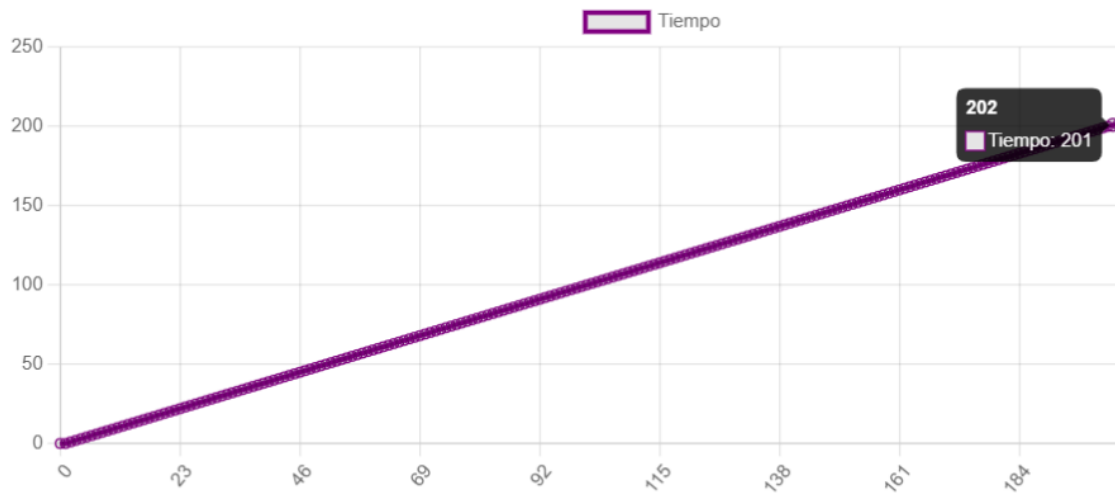


Figura 8. Tiempo tardado en apilar las cajas en mesa (frames de 5 en 5)

- **Número De Movimientos Realizados Por Todos Los Robots.**

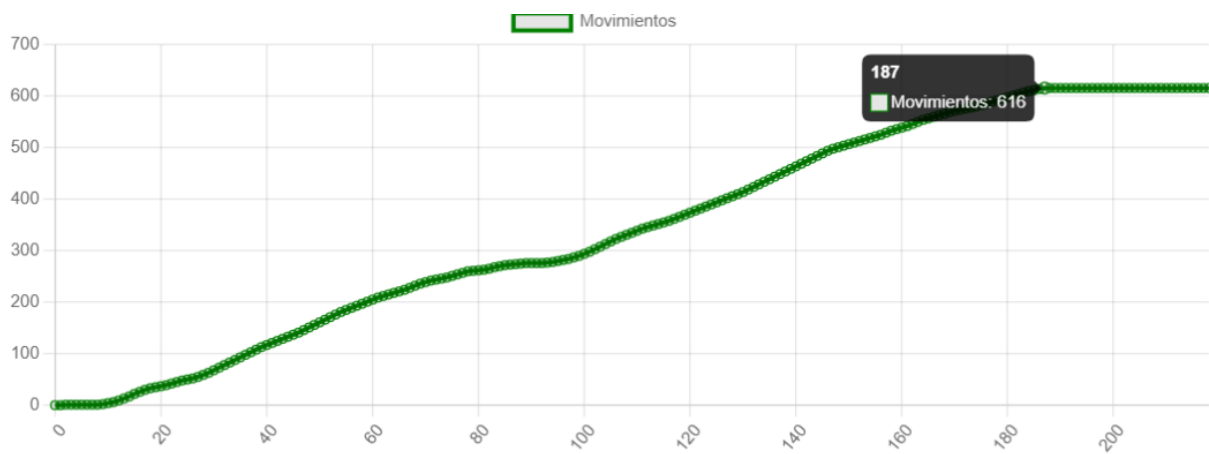


Figura 9. Movimientos de los agentes (frames de 10 en 10)

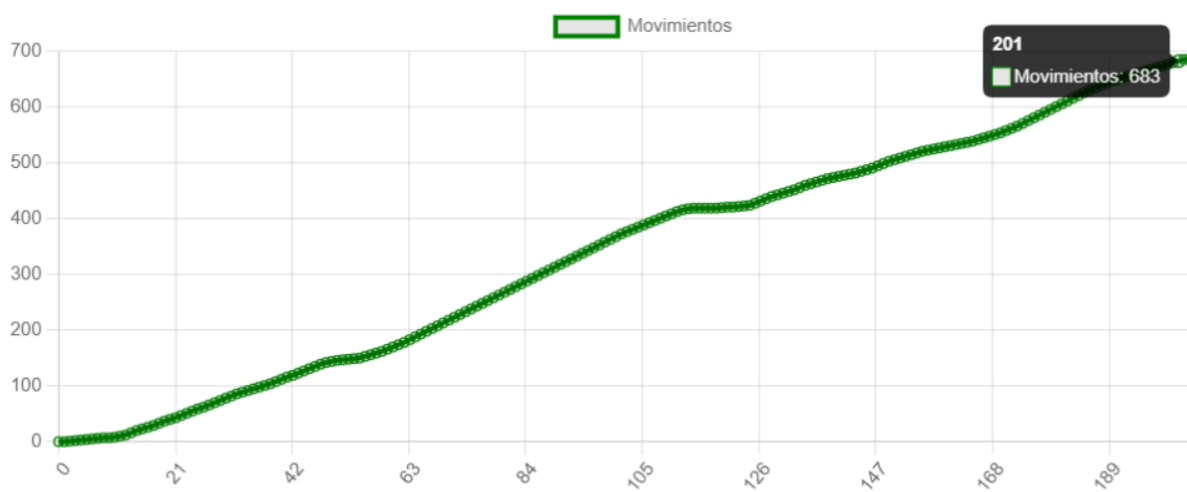


Figura 10. Movimientos de los agentes (frames de 5 en 5)

Estrategia cooperativa para la solución del problema.

Para esto se necesita de una estrategia cooperativa como los ejemplos y lo que se ha hablado en clase, se necesitaría que los robots se comuniquen entre ellos, para esto dejan una tipo marca en las celdas a las que ya fueron, haciendo así que cuando otro robot la detecte, la descarta en automático ya que ahí no hay cajas y buscaría otra para poder irse moviendo. Además, a veces los agentes se quedaban atrapados en sus casillas, para evitar esto sería necesario que los robots tuvieran un contador propio, el cual se actualizará cada vez que los demás robots dejen la marca de que ya fueron a esa celda.

También sería bueno que los robots se puedan avisar entre ellos las coordenadas de la tarima o pila en donde llevarán las cajas, haciendo que los otros robots no tomen cajas de la pila de ese y así cooperen entre ellos para optimizar el funcionamiento y el tiempo en resolver la problemática o incluso, puedan ser capaces de considerar los lugares donde podrían organizarse las pilas, los otros agentes las detecten y optimicen el tiempo en que se trasladan para apilar cajas.

Gráficas Computacionales

Modelo 2D

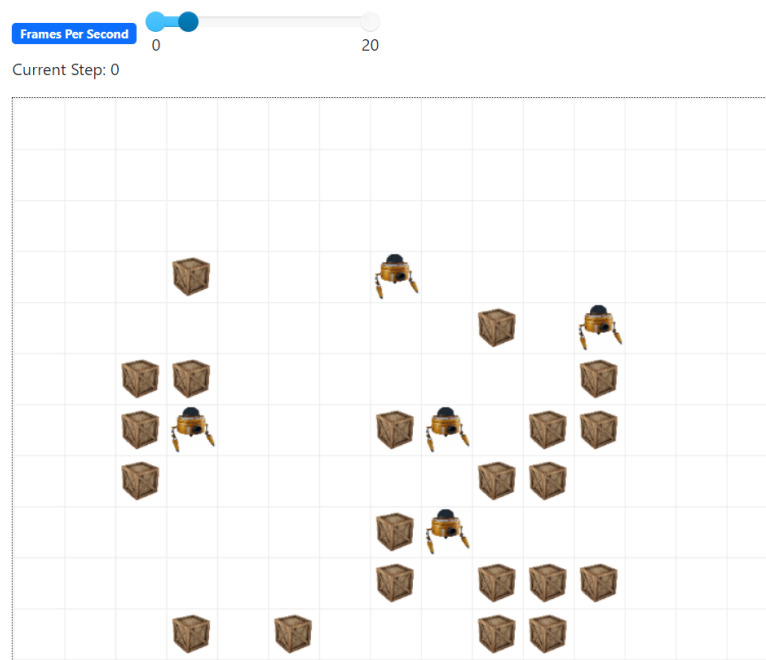


Figura 11. Agentes Buscando Cajas

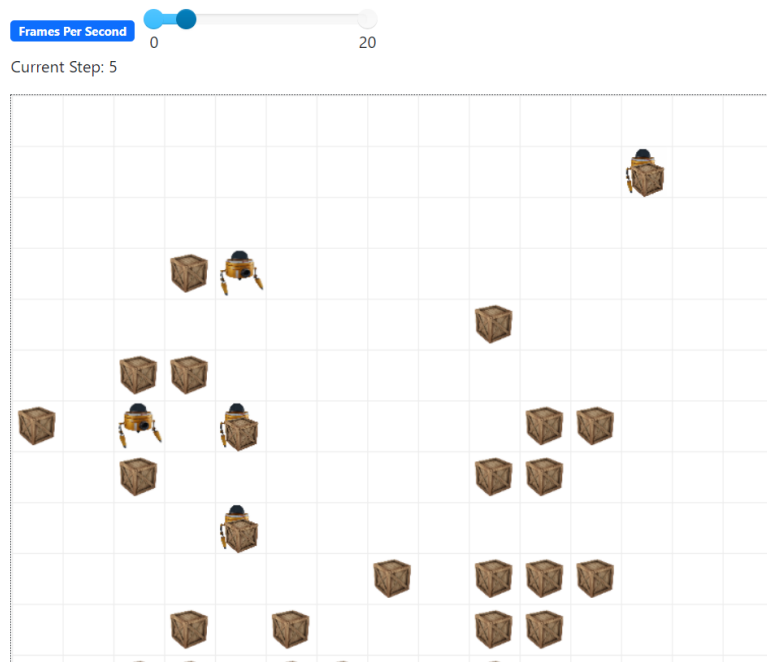


Figura 12. Agentes Cargando Cajas

Modelo 3D



Figura 13. Agentes Buscando cajas (Luz amarilla)



Figura 14. Agente llevando caja (Luz azul)

Materiales de Diseño

- Caja



Figura 15: Caja usada para el proyecto en Unity

- Robot



Figura 16: Robot usado en Unity

- Almacén

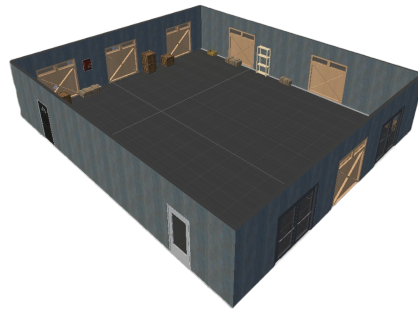


Figura 17: Ambiente en Unity

Acceso a Proyecto en Github

Proyecto Unity:

<https://github.com/SergioGonzalez24/Movilidad-Urbana-MSMGC-GPO-302/tree/main/ActividadIntegradora/Unity>

Código Servidor y mesa:

<https://github.com/SergioGonzalez24/Movilidad-Urbana-MSMGC-GPO-302/tree/main/ActividadIntegradora/Server>

.Exe escena robots:

<https://github.com/SergioGonzalez24/Movilidad-Urbana-MSMGC-GPO-302/tree/main/ActividadIntegradora/VisualizacionEscenaRobots>