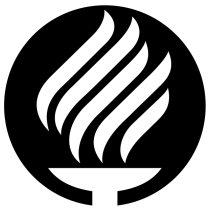


Tecnológico de Monterrey

Modelación de sistemas multiagentes con gráficas computacionales

TC2008B.523

Dr. Sergio Ruiz Loza
Dr. Christopher David Balderas Silva

Actividad Integradora

Jesús Daniel Lara Yamamoto - A01658318

Fecha de entrega: 23 de noviembre 2021

Introducción

En este documento se redacta la solución para la sección de multiagentes correspondiente a la actividad integradora, en esta, somos el propietario de 5 robots nuevos y un almacén lleno de cajas. El dueño anterior del almacén lo dejó en completo desorden, por lo que depende de nuestros robots organizar las cajas en algo parecido al orden y convertirlo en un negocio exitoso.

Es importante recordar que todos los robots están equipados con ruedas omnidireccionales y, por lo tanto, pueden conducir en las cuatro direcciones. Pueden recoger cajas en celdas de cuadrícula adyacentes con sus manipuladores, luego llevarlas a otra ubicación e incluso construir pilas de hasta cinco cajas.

Todos los robots están equipados con la tecnología de sensores más nueva que les permite recibir datos de sensores de las cuatro celdas adyacentes. Por tanto, 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. Los robots también tienen sensores de presión equipados que les indican si llevan una caja en ese momento.

Para la solución nuestro objetivo es que todos las cajas estén apiladas y de esta manera regresarle el orden a nuestro almacén. La organización de los agentes depende de ti, siempre que todas las cajas terminen en pilas ordenadas de cinco.

Solución

- Agentes:
  - Cajas
  - Robots
- Ambiente:
  - Almacén

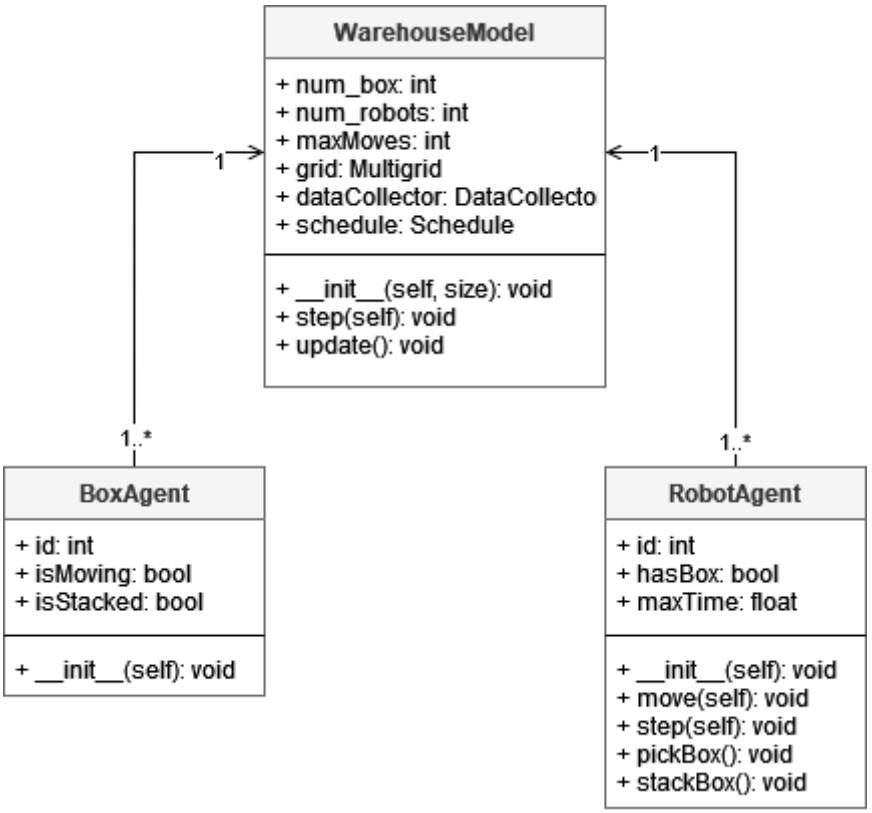
Inicializamos los agentes de caja de forma aleatoria en el ambiente donde solamente pueden estar a nivel de piso, es decir, no pueden estar apiladas (aún) e inicializamos los agentes robot de igual manera aleatoriamente pero en las casillas que están vacías. Y finalmente declaramos un tiempo máximo de ejecución.

Durante la ejecución de la simulación se recopila la siguiente información:

- El tiempo final para que todas las cajas que se encuentren en el ambiente estén apiladas en pilas máximas de 5 cajas.
- El número total de movimientos realizados por todos los robots.

Diagramas de clases

Para resolver este problema primero se planteó el diagrama de clases. Ya que primero necesitamos definir cómo serán las interacciones entre las clases. En el siguiente diagrama se muestran la clase del ambiente *WarehouseModel* y las dos clases agentes *BoxAgent* y *RobotAgent*.



Para crear la cuadrícula se usa la clase *MultiGrid* que viene incluida en la librería de Mesa, este cuadrícula permite que múltiples agentes existan en la misma celda, de esta manera podremos apilar tanto las cajas como poder tener el robot y la caja en la misma celda.

De igual forma *Schedule* y *DataCollector* provienen de las librerías Mesa, las cuales permiten que los agentes den un paso (avancen) cuando el ambiente lo requiera y activarlos de forma aleatoria y almacenar la información de cada paso (movimientos totales de los robots y tiempo final de ejecución) respectivamente.

La función __init__ es la que inicializará todos los parámetros preestablecidos en un grid de tamaño NxM, colocará los agentes en la cuadrícula en los espacios establecidos, asignará un tiempo máximo de ejecución y guardará información de cada iteración.

La función update nos ayudará a actualizar la función step con respecto al tiempo máximo de ejecución.

Por otro lado, la clase *RobotAgent* será la encargada de ser uno de los agentes simulados por el ambiente, la cual contiene un booleano que nos permitirá saber si el robot tiene una caja actualmente y un flotante para saber el tiempo máximo de ejecución del agente.

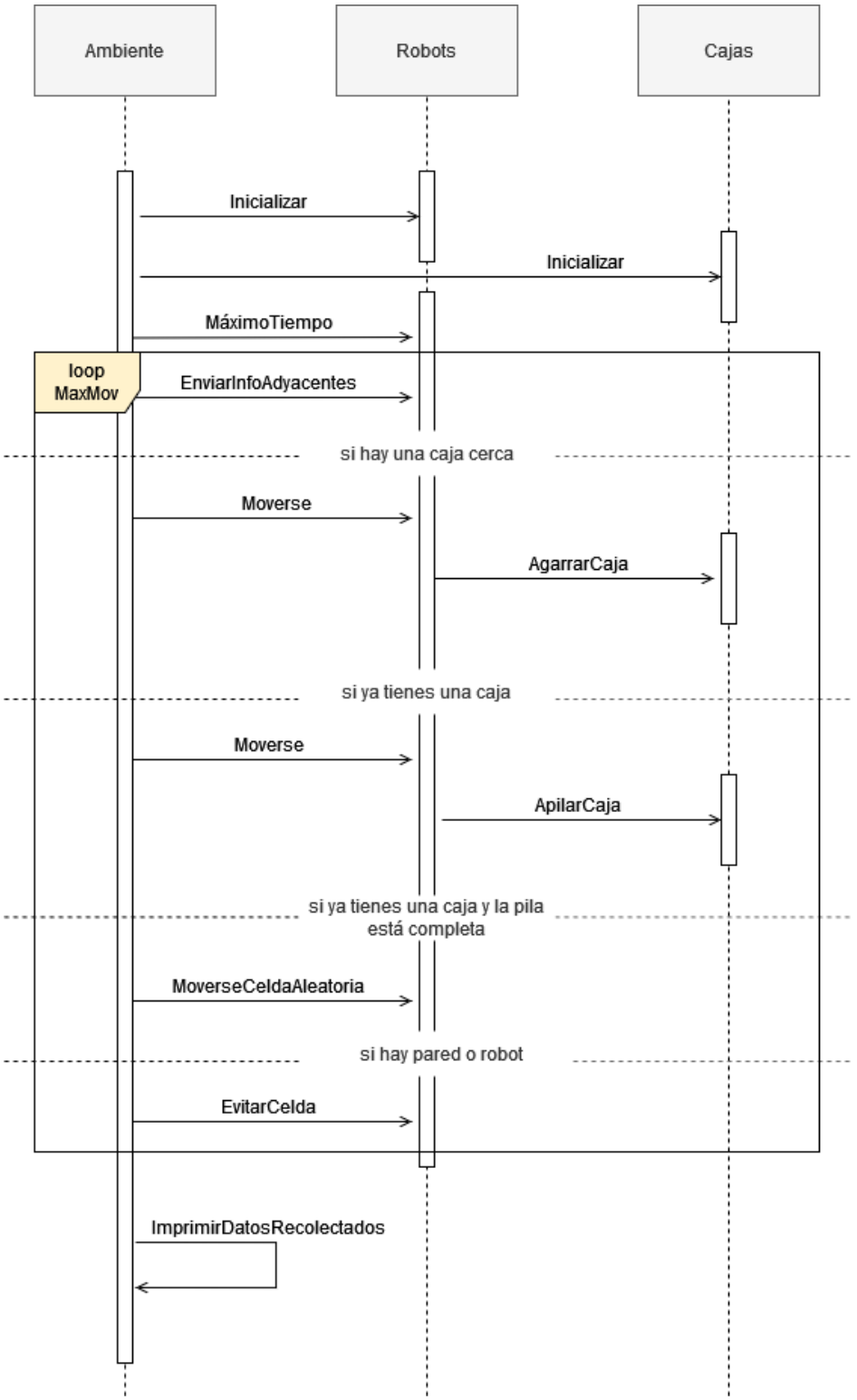
Esta clase cuenta con distintos métodos como agarrar y apilar caja, moverse y paso, las cuales serán llamadas por el ambiente en cada paso. Además, cuenta también con su propia función __init__ para inicializar las variables del agente cuando este sea creado.

Por último, la clase *BoxAgent* es el otro agente simulado por el ambiente, la cual contiene dos booleanos, *isMoving* y *isStacked* para saber cuando la caja está en movimiento o está apilada con otras cajas.

Esta clase solo cuenta con la función __init__ para inicializar los valores del agente cuando sea creado por el ambiente.

Diagrama de protocolos de interacción

Posteriormente, después de hacer el diagrama de clase se planteó el diagrama de protocolos de interacción para tener una idea de cómo será el proceso que seguirá el programa durante la simulación. En la siguiente figura se muestra el diagrama.



Lo primero que hará el programa será crear el tablero para poder inicializar los agentes (robots y cajas), posteriormente se definirá el tiempo máximo de ejecución por robot y los movimientos máximos que habrá en la simulación.

Una vez definido el número de movimiento máximos se hará un ciclo de los posibles movimientos que pueden hacer los robots, ya que estos agentes pueden recoger una caja pero si están cargando una caja tiene que apilarla en otra siempre y cuando dicha pila no esté completa, pero también está la posibilidad de encontrarse con otro agente o una pared.

Finalmente, después de acabar el ciclo el ambiente desplegará toda la información recopilada de los pasos de todos los agentes, es decir, mostrará el total de pasos de todos los agentes y el tiempo total de la ejecución.

Propuesta para optimizar el número de movimientos

Para este tipo de problemáticas podemos buscar alternativas para optimizar factores como el tiempo de ejecución y los movimientos totales realizados por los robots. Por ejemplo, podemos asignarles cuadrantes específicos a cada robot. Cada robot se encargará de apilar las cajas existentes en su cuadrante y de esta manera los robots no perderán tiempo en desplazarse a un cuadrante más lejano en busca de una caja.

La implementación de esta solución reducirá el tiempo de ejecución de la simulación, además de minimizar los movimientos de los robots, al reducir el espacio en el que cada uno de ellos se desplazara.