

**AI. Actividad Integradora**

**Materia**

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 4)

**Alumnos**

Armando Montaño González - A00827377

**Profesores**

Edgar Covantes Osuna

Jorge Mario Cruz Duarte

13 de Agosto del 2021, Monterrey N.L.

**Introducción**

En este documento se implementará el modelado y planteará la simulación de un almacén con K cajas y N robots que deberán recogerlas y llevarlas a un estante en una posición arbitraría, el cual tendrá cupo solo para 5 cajas. Se analizará una posible solución, y se diseñarán los agentes involucrados, asi como el modelo. Por último, se abarcará un poco en las posibles áreas de mejoras y fallos que tiene la solución actual.

**Definición del Problema**

Felicidades! Eres el orgulloso 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 tus robots organizar las cajas en algo parecido al orden y convertirlo en un negocio exitoso. Cada robot está equipado con ruedas omnidireccionales y, por lo tanto, puede 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. Lamentablemente, tu presupuesto resultó insuficiente para adquirir un software de gestión de agentes múltiples de última generación. Pero eso no debería ser un gran problema ... ¿verdad? Tu tarea es enseñar a sus robots cómo ordenar su almacén. La organización de los agentes depende de ti, siempre que todas las cajas terminen en pilas ordenadas de cinco.

* Realiza la siguiente 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 posición aleatorias vacías.
  + Se ejecuta el tiempo máximo establecido
* Deberás recopilar la siguiente información durante la ejecución:
* Tiempo necesario hasta que todas las cajas están en pilas de máximo 5 cajas
* Número de movimientos realizados por todos los robots.
* Analiza si existe una estrategia que podría disminuir el tiempo dedicado, así como la cantidad de movimientos realizados. ¿Cómo sería? Descríbela

**Agentes Involucrados y su tipo**

Para esta simulación es completamente necesaria la participación directa de al menos 2 Agentes: Robots, y Piso. Estos dos agentes interactuarán entre ellos y podrán mantener una comunicación simple.

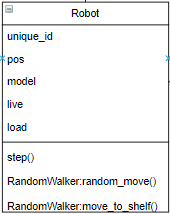
Para saber de qué estarán “compuestos” dichos agentes y como se comportarán, hay que entender un poco la problemática:

* Se nos indica que es un almacén
* Son robots que se mueven ARRIBA, ABAJO, IZQUIERDA O DERECHA
* No deben de estar en la misma celda los robots
* Recogen cajas y las llevan a un estante en una posición designada

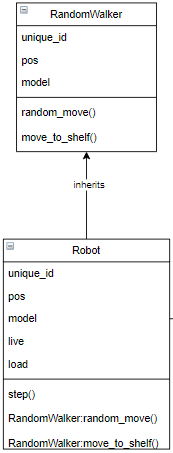
Teniendo esto en cuenta podemos definir ciertas cosas para cada agente:

1. Robot

* Tendrá un unique\_id
* El model en el que se encuentra
* La posición actual del robot
* Su estado (si existe o no)
* Si tiene carga o no (load)
* A pesar de que lo ideal es que se un agente que aprenda, el tipo Robots se planea que sea Agente reactivo simple
* Un ejemplo de la clase Agente Robot podría verse de la siguiente manera:

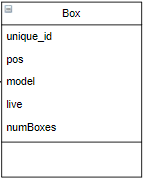


Cabe destacar que la Clase Robot hereda métodos de otra clase llamada RandomWalker (la cual simplemente está ahí para hacer el movimiento posible, nada más), una diagramación más acertada podría verse así:

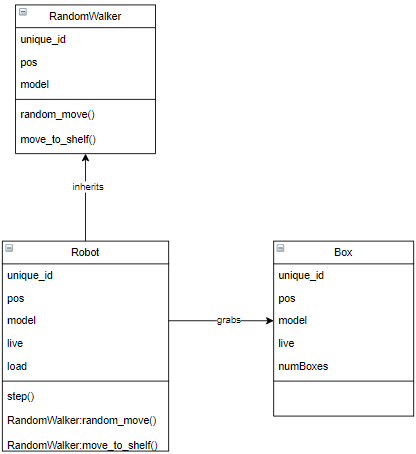


1. Box

* Tendrá un id único
* El modelo en el que se encuentra
* La posición de la caja
* El estado de la caja (si hay una caja ahí o no)
* El número de cajas que hay en ese lugar
* Su tipo será Agente Reactivo Simple, este no tiene por qué aprender nada dado que se supone que es un objeto inerte
* Un ejemplo de la clase Agente Semáforo podría verse de la siguiente manera:



En conjunto, podrían verse así:



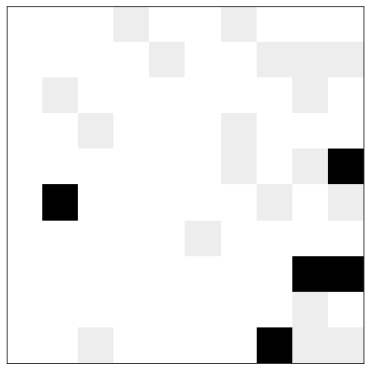
**Clase modelo**

Es una cuadricula MxN, que tiene que actuar como almacén, habrá que definir donde estarán los espacios designados como estantes, y en donde estarán posicionados las cajas y los robots. A su vez, hay que definir ciertas cosas previamente:

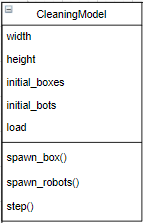
* El tamaño del ‘Grid’ puede variar (NO durante la simulación, claramente)
* Se designará como primer estante la posición (0,0), si se llena será la (1,0), posteriormente la (2,0), es decir, la posición si se llena un estante será (i+1,0).
* El número de robots y cajas iniciales puede ser una variable, y no un número fijo

Finalmente, lo que tendrá que hacer el modelo correctamente es:

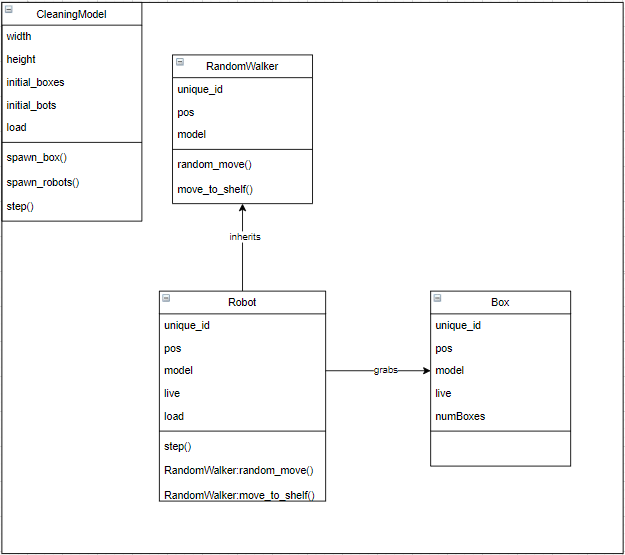
* Inicializar y posicionar correctamente los robots y las cajas sin que aparezcan en el mismo lugar (nada debe de estar en la misma posición al spawnear)
* Designar el espacio del estante
* El tiempo de ejecución y número de generaciones



Un ejemplo de cómo se podría ver lo que hace la clase modelo es el siguiente:



En conjunto se vería de la siguiente manera:



**Parametrización**

Aquí hablaré sobre los distintos parámetros con los que podríamos jugar/modificar para ver cómo se comporta la simulación en distintos casos del mismo escenario.

Dada la naturaleza del problema, el único parámetro importante que podría modificarse es el número de cajas y bots iniciales que existen en el modelo, esto podría ser una variable que podemos cambiar cada vez que queramos correr la simulación. Aunque también podríamos modificar el número de generaciones o el tiempo máximo de ejecución para que actuase como una prueba contra tiempo/generaciones.

**Posibles Mejoras**

La principal y más grande mejora que se podría hacer es el movimiento de los bots. Al momento de hacer esto el movimiento es completamente al azar, de tal manera que es literalmente la peor manera posible y bruta de hacerlo. Posibles mejoras en cuanto a esto serían las siguientes:

* Que el movimiento siga siendo random (dado que es muy fácil de implementar y no quita nada de tiempo), pero que cuando tenga una caja a su alrededor (get\_neighborhood) se dirija hacia allí en vez de seguir moviéndose aleatoriamente
* Implementar Dijkstra a los bots o algún algoritmo de ‘Find the Shortest path’ a los robots. Aunque puede tomar más tiempo
* Implementar la misma lógica que se usó para ir al estante, pero ahora con cada caja.
* Que los bots pasen por cada celda del grid. Es decir, que donde spawneen, empiecen a barrer la zona hacia la derecha (por ejemplo), luego abajo, luego izquierda, etc. Se podría ver así:



Otra posible mejora es hacer que donde encuentren la caja, ahí se haga estante, reduciendo así los movimientos que tienen que hacer para dirigirse a la posición designada. Aunque de esta manera romperíamos un poco con la condición de ordenar, dado que simplemente estarían apilando cajas en el lugar que la encontraron y el almacén no quedaría “ordenado” exactamente.