

Valeria Pineda González A01023979

Edgar Ivan Rostro Morales A01029036

Eduardo Villalpando Mello A01023646

TC2008B Modelación de Sistemas Multiagentes con Gráficas Computacionales

Prof. Gilberto Echeverría

Prof. Octavio Navarro

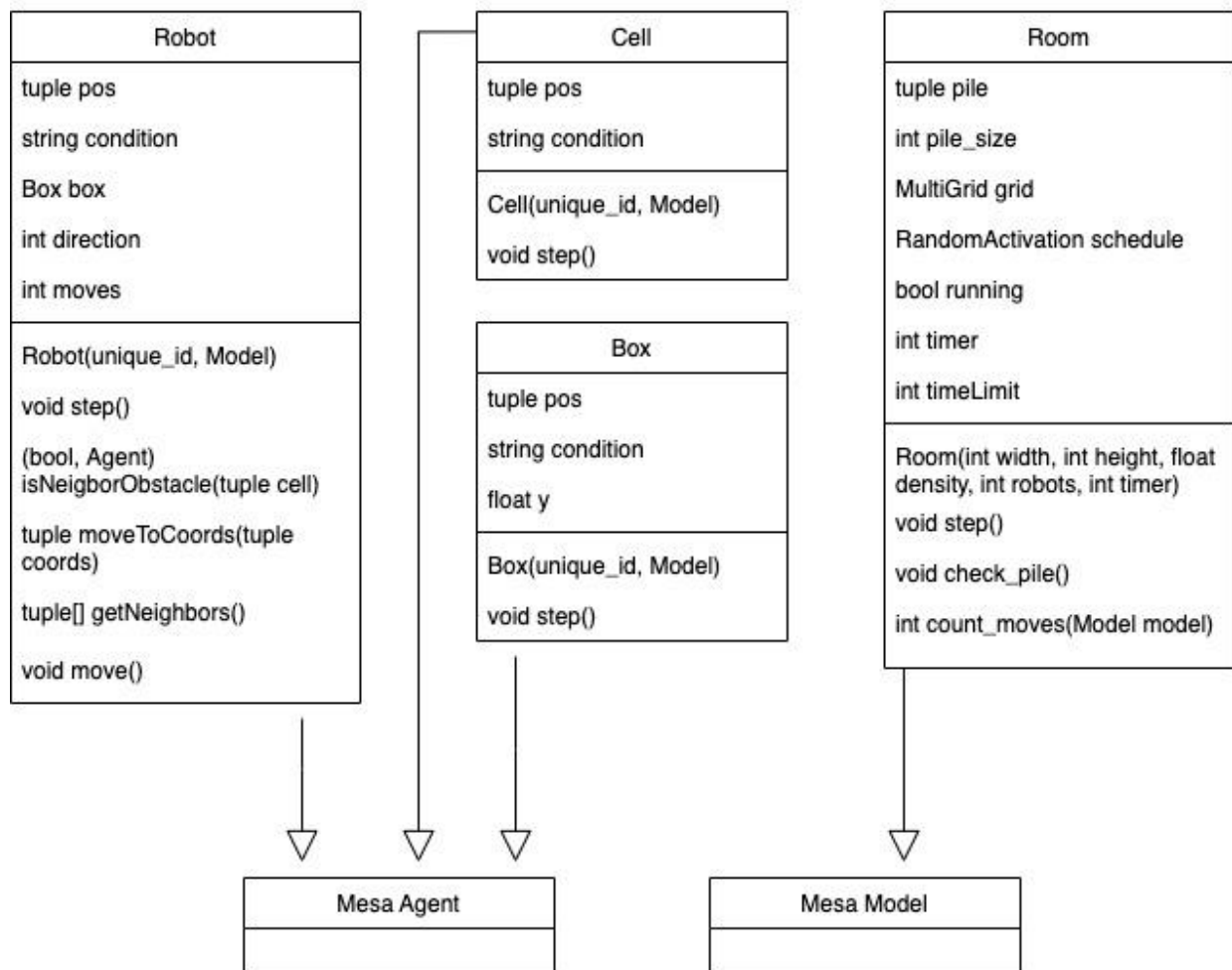
24 de noviembre de 2021

## Actividad Integradora: Almacén

### Objetivo

Crear un sistema multiagentes, donde una cantidad  $k$  de robots acomoda todas las cajas (en posiciones iniciales aleatorias) en pilas de 5 en la esquina inferior izquierda del almacén.

### Diseño



**Figura 1.** Diagrama de clases de objetos del almacén.

## ***Robots***

### Estados

- Buscando cajita
- Acomodando cajita

### Protocolos (ordenadas por prioridad)

- Si está buscando cajita, el robot debe dirigirse a la cajita vecina no acomodada más cercana.
- Si está buscando cajita y no tiene cajita en alguna celda vecina, se mueve aleatoriamente.
- Si está acomodando cajita, el robot debe dirigirse a la posición para acomodarla usando la ruta más corta: primero moviéndose en el eje donde la distancia sea mayor.
- Si está acomodando caja y se encuentra con algún obstáculo (otro robot u otra caja), debe rodearla y evitar el obstáculo

## ***Cajitas***

### Estados

- No acomodada
- En movimiento
- Apilada

### Reglas

- Si está en acomodamiento su posición es la misma que la caja (más arriba en el eje y)
- Empezar cajas como apiladas, para posteriormente posicionar cajas sobre esa caja

## ***Celdas***

### Estados

- Disponible
- Obstáculo

### Reglas

- Si tres de sus vecinos son obstáculos, cambiar estado a *obstáculo*.

## **Resultados**

### Variables independientes

- Densidad de cajas (default: 0.2)

- Tamaño del grid (default: 10x10)
- Cantidad de robots (default: 3)
- Tiempo límite (default: 100s)

#### Variable dependientes

- Tiempo transcurrido
- Cantidad de movimientos
- Cantidad de cajas apiladas

<b>Tabla 1. Pruebas de densidad de cajas</b>			
<b>Densidad</b>	<b>Tiempo (s)</b>	<b>Movimientos</b>	<b>Cajas apiladas (%)</b>
0.1	100	115	36
0.2	100	118	22
0.3	100	144	0

<b>Tabla 2. Pruebas de tamaño del cuarto</b>			
<b>Tamaño del grid</b>	<b>Tiempo (s)</b>	<b>Movimientos</b>	<b>Cajas apiladas (%)</b>
5 x 5	48	51	100
10 x 10	100	116	22
15 x 15	100	113	13

<b>Tabla 3. Pruebas de cantidad de robots</b>			
<b>Cantidad de robots</b>	<b>Tiempo (s)</b>	<b>Movimientos</b>	<b>Cajas apiladas (%)</b>
1	100	36	33
2	100	76	21
3	100	110	35

<b>Tabla 4. Prueba de tiempo límite</b>			
<b>Tiempo límite (s)</b>	<b>Tiempo (s)</b>	<b>Movimientos</b>	<b>Cajas apiladas (%)</b>

50	50	58	13
100	100	112	29
150	150	159	50

### Propuestas de mejora

- Enviar mensaje de ayuda a los otros robots en caso de que un robot quede atrapado entre obstáculos para que éstos puedan recogerlos.
- Si está viajando con una caja y se encuentra con otra en su camino, dejar la que estaba llevando y recoger la que le estorba.
- Planear una ruta a seguir por cada robot, evitando invadir rutas ajenas y buscando cajas.
- Usar un algoritmo eficiente para encontrar las coordenadas de la caja más cercana.
- Marcar posición de caja vecina para que otros agentes sepan que ahí hay una caja.
- Incrementar la capacidad de carga del robot para transportar varias cajas y evitar múltiples recorridos.
- Tener múltiples zonas de descarga para evitar rutas ineficientes.
- Cuando se lleva una caja, calcular una ruta al lugar de descarga, enviar esta ruta a los demás robots para que recalculen su ruta evitando colisiones.
- Definir una zona de descarga que sea suficiente para acomodar las cajas existentes.
- Empezar a recoger las cajas que estén la zona de descarga definida.

### Problemas encontrados

- Muchas veces los robots pueden quedar atrapados al estar rodeados de cajas.
- La librería Mesa puede dejar de funcionar inesperadamente.