

# TC2008B. Modelación de Sistemas Multiagentes con Gráficas Computacionales

## Actividad Integradora

### Parte 1. Sistemas multiagentes

#### Descripción 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.

## Parte 2. Gráficas Computacionales

### Descripción del problema

Aplica la misma descripción de la actividad en la Parte 1.

Tu trabajo consiste en modelar y desplegar la representación en 3D del mismo. El diseño y despliegue debe incluir:

- Modelos con materiales (colores) y texturas (usando mapeo UV):
  - Estante (con repetición de instancias o prefabs por código).
  - Caja (con repetición de instancias o prefabs por código).
  - Robot (con repetición de instancias o prefabs por código, al menos 5 robots).
  - Almacén (piso, paredes y puerta).
- Animación
  - Los robots deberán desplazarse sobre el piso del almacén, en los pasillos que forman los estantes.
  - Para esta actividad, no es necesario conectar la simulación con el despliegue.
- Iluminación
  - Al menos una fuente de luz direccional.
  - Al menos una fuente de luz puntual sobre cada robot (tipo sirena). Dicha luz se moverá con cada robot.
- Detección de colisiones básica
  - Los robots se moverán en rutas predeterminadas.
  - Los robots se moverán con velocidad predeterminada (aleatoria).
  - Los robots comenzarán a operar en posiciones predeterminadas (aleatorias).
  - Los robots detectarán y reaccionarán a colisiones entre ellos. Determina e implementa un sistema básico para esto (por ejemplo, detenerse previo a una colisión y asignar el paso a uno de los robots).

### Especificaciones de entrega

Deberás entregar el **enlace a un repositorio personal de GitHub** que contenga:

- Para la Parte 1, un documento PDF con los diagramas de clases y protocolo de agentes, estrategia cooperativa para la solución del problema.
- Para la Parte 1, el código implementado para la simulación del reto.
- Para la Parte 2, el archivo **.unitypackage** con todo lo necesario para ejecutar la solución.

## Notas

Para ambas partes del problema, considera que:

- Tienes un almacén de  $M \times N$  espacios.
- $K$  cajas iniciales, en posiciones aleatorias.
- 5 robots.
- Tiempo máximo de ejecución.