

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 objetos. El dueño anterior del almacén lo dejó en completo desorden, por lo que depende de tus robots organizar los objetos en algo parecido al orden y convertirlo en un negocio exitoso.

Cada robot está equipado con un sistema de tracción potente y, por lo tanto, puede avanzar sobre cualquier terreno, y también girar en cualquier dirección. Pueden recoger objetos en celdas de cuadrícula frente a ellos con sus manipuladores, luego llevarlas a otra ubicación e incluso construir pilas de hasta cinco objetos. 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 objetos (y cuantos objetos hay en la pila) o está ocupado por otro robot. Los robots también tienen sensores de presión equipados que les indican si llevan un objeto en ese momento. Estos Robots también tienen capacidad de llevar un mapa consigo.

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 todos los objetos terminen en pilas ordenadas de cinco.

- Realiza la siguiente simulación:
 - Inicializa las posiciones iniciales de los K objetos. Todos los objetos están a nivel de piso, es decir, no hay pilas de objetos.
 - Todos los agentes empiezan en posición aleatorias vacías.
 - Se ejecuta el tiempo máximo establecido (o steps máximos).
- Deberás recopilar la siguiente información durante la ejecución:
 - Tiempo necesario hasta que todos los objetos están en pilas de máximo 5 objetos.
 - Número de movimientos realizados por cada robot.
 - Analiza si existe una estrategia que podría disminuir el tiempo dedicado, así como la cantidad de movimientos realizados. ¿Cómo sería? Descríbela.
- Puntos para tomar en cuenta:
 - Los robots solo pueden operar con razonamiento deductivo, razonamiento

práctico, o una combinación de las dos

- Los robots detectarán colisiones entre ellos en el mundo 3D. Determina e implementa un sistema básico para esto (por ejemplo, detenerse previo a una colisión y asignar el paso a uno de los robots).
- Los robots solo pueden avanzar hacia el frente, pero pueden girar a todas las direcciones sobre su propio eje.
- Deberás diseñar, implementar y usar una ontología en cada agente.

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).
 - Objetos varios (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.
- 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 colisiones entre ellos.

Parte 3. Visión Computacional

Descripción del problema

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

Tu trabajo consiste identificar los objetos que se están llevando a apilar:

- Asume que el robot tiene una cámara
 - Puede ser hacia el frente o cómo creas conveniente.
 - La cámara hace un stream hacia el modelo de visión
- Puedes usar cualquier modelo de visión
 - SAM, YOLO, OpenAI, etc.
- Cuando el robot encuentra un objeto (lo percibe), debe dar alguna señal de qué es lo que está observando (e.g. un globo de diálogo que dice: “¡Es una manzana!”)

Especificaciones de entrega

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

- Para la Parte 1, un documento PDF con especificaciones de Propiedades de agentes y ambiente, así como una métrica de utilidad o éxito de cada agente. También deben estar los diagramas de clases de los agentes utilizados, y diagramas de clase de las ontologías utilizadas. Finalmente, agrega una conclusión donde se incluya una breve descripción de soluciones alternativas para mejorar la eficiencia de los agentes.
- 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 el problema, considera que:

- Tienes un almacén de $M \times N$ espacios.
- K objetos iniciales, en posiciones aleatorias.
- Al menos 5 robots.
- Tiempo máximo de ejecución (segundos o steps).