



Tecnológico de Monterrey

Curso:

Modelación de sistemas multiagentes con gráficas computacionales

Grupo:

302

Estudiante:

Juan de Dios Gastélum Flores - A01784523

Maestro:

Gilberto Echeverría Furió

Octavio Navarro Hinojosa

Título:

MA. Actividad: Roomba

Fecha de entrega:

19 de Noviembre 2025

Introducción y Planteamiento del Problema

El objetivo de esta actividad es diseñar un sistema de agentes que limpien una habitación de tamaño $M \times N$, la cual contiene suciedad y obstáculos colocados aleatoriamente.

Cada agente cuenta con batería limitada y debe regresar a una estación de carga antes de agotarla. El reto principal es equilibrar limpieza, movimiento eficiente y supervivencia energética.

Análisis y Diseño de Agentes

Características del Agente

Objetivo: maximizar el porcentaje de celdas limpias dentro del tiempo máximo de ejecución.

- Sensores:
 - Nivel interno de batería.
 - Estado de la celda actual (sucia o limpia).
 - Detección de estaciones de carga en la posición actual.
 - Celdas vecinas libres de obstáculos.

- Actuadores:
 - Movimiento a celdas vecinas (costo: 1% batería).
 - Limpieza de la celda actual (1% batería).
 - Recarga en estación de carga (5% por paso).

Arquitectura de Subsunción

El comportamiento del agente se organiza por prioridades:

- Supervivencia (Alta):
 - a. Si la batería es baja (<20%), regresa a la estación de carga siguiendo la ruta más corta.
 - b. Si está en estación y no está lleno, recarga.

- Trabajo (Media):
 - a. Si la celda actual está sucia: limpiar inmediatamente.
- Exploración (Baja):
 - a. Si no hay tareas urgentes, se desplaza para buscar nuevas celdas sucias.

Navegación

Para regresar a la base, el agente usa Dijkstra, lo que garantiza encontrar rutas válidas evitando obstáculos.

Análisis del Ambiente

El entorno es una cuadrícula con:

- Distribución aleatoria de suciedad.
- Obstáculos impasables.
- Estaciones de carga.
- Cambios dinámicos conforme los agentes limpian y se mueven.

Simulación 1: Un agente

Un solo agente inicia con una única estación de carga.

Simulación 2: Varios agentes

Se colocan N agentes en posiciones aleatorias, y cada uno inicia con su propia estación de carga. Los agentes pueden usar cualquier estación disponible, lo que reduce tiempos muertos de recarga.

Resultados

Las simulaciones se realizaron en un ambiente de 15 x 15.

Métricas	1 Agente	5 Agentes
Tiempo (steps)	Alto (>800)	Bajo (<300)
% Limpieza	Variable	100%

Observaciones:

- El sistema multiagente limpia mucho más rápido gracias a la distribución de estaciones de carga.
- El agente individual pierde tiempo viajando largas distancias para recargarse.
- Varios agentes cubren más área y reducen la probabilidad de quedarse sin batería.

Conclusiones

- La arquitectura de subsunción funcionó correctamente al priorizar supervivencia, limpieza y exploración de manera simple pero efectiva.
- El uso de Dijkstra fue clave para evitar que los agentes quedaran sin energía al regresar a la base.
- El enfoque multiagente escala de forma eficiente, reduciendo el tiempo total de limpieza y aumentando el porcentaje final de celdas limpias.
- En espacios pequeños, demasiados agentes pueden estorbarse, pero en ambientes medianos y grandes el rendimiento mejora significativamente.