



Tecnológico de Monterrey

Campus Santa Fe

Modelación de sistemas multiagentes con gráficas computacionales.

TC2008B.302

Actividad:

Roomba

Alumnos:

David Santiago Vieyra García - A01656030

Profesor:

Octavio Navarro Hinojosa

Fecha de entrega:

16 de noviembre de 2023

1. Introducción

Se realiza una simulación para abordar la problemática de la limpieza de un entorno mediante agentes Roomba, diseñados para realizar tareas de limpieza, y gestión de la energía en un espacio simulado. El propósito es evaluar la eficacia y eficiencia de estos agentes en la realización de sus tareas asignadas.

Para este proyecto de simulación se utilizó el lenguaje Python y la librería Mesa, se aborda la implementación de agentes Roomba, agentes de obstáculos, agentes de suciedad y agentes de estaciones de carga para llevar a cabo tareas de limpieza en un entorno simulado. El proyecto se estructura en tres archivos principales:

- *agent.py*: Definición de las clases para los agentes Roomba (encargados de la limpieza), obstáculos, suciedad, estaciones y celdas visitadas.
- *model.py*: Inicialización del entorno, asignación de posiciones iniciales a los agentes, y distribución aleatoria de obstáculos y suciedad en el espacio simulado.
- *server.py*: Creación de la interfaz de usuario mediante CanvasGrid de Mesa, configuración de sliders para ajustar parámetros y gráficos para visualizar estadísticas en tiempo real.

2. Análisis de Agentes

Objetivo de los Agentes Roomba

Los agentes Roomba tienen como objetivo principal limpiar el entorno simulado y mantener una carga adecuada de su batería para continuar con la limpieza en el menor tiempo posible.

Capacidades Efectoras y de Percepción

Los agentes Roomba tienen la capacidad de moverse en 4 direcciones (adelante, atrás, derecha e izquierda), identificar celdas sucias, calcular la ruta más corta a su estación de carga, recolectar basura y evitar obstáculos. Su percepción se basa en la detección de obstáculos, presencia de agentes basura, otros roomba, nivel de batería y conocer la ubicación de su estación de carga.

Arquitectura de Subsunción Utilizada

La arquitectura de subsunción empleada permite priorizar las acciones de los agentes roomba, garantizando la gestión de la batería y evitar obstáculos como objetivos primordiales, seguidos de la limpieza de la celda dónde se encuentra y el movimiento a celdas vecinas.

Comportamiento Prioritario - Regresar a la Estación de Carga si la Batería está Baja

- Condición de Activación: Nivel de batería por debajo del umbral. Acción: Cambiar el objetivo a "Regresar a la Estación de Carga" y dirigirse hacia ella.

Comportamiento - Evitar Obstáculos

- Condición de Activación: Detección de obstáculos en el radio de percepción.
- Acción: Cambiar la dirección del movimiento para evitar el obstáculo.

Comportamiento - Limpiar

- Condición de Activación: Detección de celdas sucias en el radio de percepción.
- Acción: Limpiar la celda actual.

Comportamiento - Movimiento General

- Condición de Activación: Cuando no se activan comportamientos prioritarios.
- Acción: Moverse en una dirección aleatoria para explorar el entorno y buscar áreas no limpiadas.

Métricas de Desempeño

Las métricas utilizadas incluyen el porcentaje de celdas limpiadas, el porcentaje de celdas visitadas y la eficiencia en la gestión del tiempo durante la limpieza.

3. Diseño del Ambiente

El entorno simulado consta de un espacio bidimensional con obstáculos y agentes de basura distribuidos aleatoriamente, estaciones de carga y el agente Roomba. El espacio se divide en celdas, y la interacción ocurre entre los agentes y estos espacios.

4. Experimentos y Estadísticas Recolectadas

Se realizaron múltiples simulaciones variando las condiciones iniciales, como la distribución de obstáculos y la cantidad de basura. Las estadísticas recolectadas incluyen el número de pasos dados por los agentes y porcentaje de celdas limpiadas.

5. Análisis de Resultados y Conclusiones

Resultados

Se observó que los agentes Roomba lograron limpiar un alto porcentaje de celdas. Sin embargo, se identificó una necesidad de mejora en la eficiencia al navegar al punto de carga y al desplazarse en el ambiente.

Conclusiones

El desempeño de los agentes Roomba en ocasiones fue efectivo en la limpieza, pero se sugiere investigar y mejorar los algoritmos para perfeccionar la estrategia del recorrido para optimizar el uso de la energía.

6. Conclusiones Finales

La simulación demostró la efectividad de los agentes Roomba en la limpieza del entorno. Se reconoce la necesidad de refinamiento en las estrategias de recarga y desplazamiento para optimizar su rendimiento.