



Tecnológico de Monterrey

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 301)

Roomba

Alumno:

Angel Adrian Afonso Castellanos - A01782545

Santa Fe

2023

Problema que se está resolviendo, y la propuesta de solución.

El problema que se está abordando es la simulación de un entorno donde agentes con movimientos aleatorios deben moverse, recolectar basura y regresar a un punto de carga para recargar su energía, teniendo como objetivo maximizar la recolección de basura y gestionar eficientemente la energía de los agentes. En este trabajo se propone un modelo basado en agentes donde cada agente que no sea basura u obstáculo toma decisiones para moverse, recolectar basura y recargar energía. Además, se implementó la capacidad de recargar energía en un punto de carga específico (Battery Agent en la posición 1,1) cuando la vida del agente está por debajo de 20 unidades.

El diseño de los agentes (objetivo, capacidad efectora, percepción, proactividad, métricas de desempeño, etc.).

Hay cuatro tipos de agentes en la simulación y estas son algunas características de cada uno:

- Agente Limpiador: Este es el agente principal de todo el modelo, es el encargado de moverse en cualquier dirección aleatoriamente para explorar el mapa, si uno de sus posibles movimientos tiene un agente basura en la celda, este tomará preferencia por el y será removido, simulando la limpieza, de igual forma, evita obstáculos, los cuales son círculos de color gris y tiene un regreso un poco violento a su punto de origen (BatteryAgent), el cual se encargará de regenerarle la vida cuando este llegue a 21, ya que cada movimiento que realice el agente, implicara la baja de una unidad a su vida (tiene una probabilidad muy baja de 10% a no moverse y no consumir energía).
- Agente Basura: Este agente es un círculo negro generado aleatoriamente en la simulación, en cualquier celda que no sea la de la posición inicial del agente limpiador o algún obstáculo, estos no se mueven ni tienen vida, solo existen en el ambiente esperando a ser removidos por el agente limpiador. Este agente no recibe información por lo que nunca sabe que tiene a los lados. En el programa entregado se plantean 40 agentes basura a ser removidos.
- Agente Obstáculo: Este agente igual que el agente basura sólo existe en el modelo sin realizar ningún movimiento, apareciendo en cualquier parte del mapa de manera aleatoria. Su único objetivo es estorbar al agente limpiador y no tiene percepción de los agentes que tiene a su alrededor. En la simulación se colocan 20 obstáculos.
- Agente Batería: Este agente será el encargado de regenerar la vida del agente limpiador, cuando se encuentre debajo de un agente limpiador este le pondrá la vida a 100, tampoco tienen idea de que tiene a su alrededor y solo aparece al inicio de la simulación en la misma posición que el agente limpiador, simulando así un checkpoint.

La arquitectura de subsunción de los agentes.

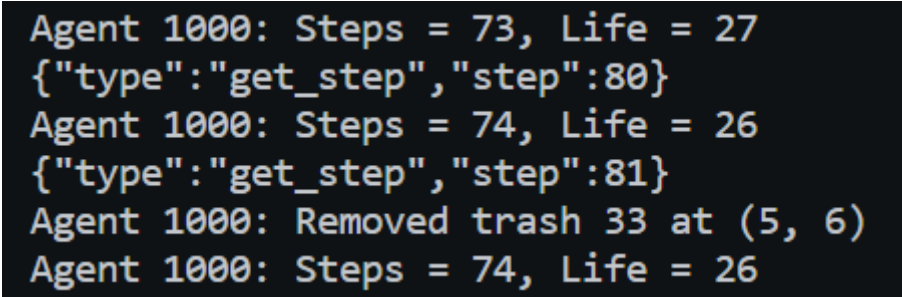
Se ha implementado una arquitectura basada en eventos donde los agentes limpiadores responden a eventos específicos, como la necesidad de recargar energía cuando la vida es baja. La lógica de subsunción se basa en las prioridades de las acciones: regresar al punto de carga si la vida es baja, recolectar basura si está disponible en cualquiera de las posiciones vecinas y moverse aleatoriamente para descubrir el entorno. Lastimosamente por falta de tiempo del programador (yo), no fue posible hacer que el agente recordara por que casillas ya había pasado, para así evitar a toda costa la repetición de las mismas y recorrer en su totalidad el mapa.

Características del ambiente.

El entorno es un grid de 10x10 casillas donde los agentes serán colocados dependiendo de su tipo. Contiene obstáculos, basura y un punto de carga. El agente limpiador navegará por el entorno, evitando obstáculos, recolectando basura y recargando energía en el punto de carga hasta que ocurra algún bug.

Las estadísticas recolectadas en las simulaciones.

Originalmente se tenía la idea de generar un mapa que indicara todas las celdas por las que había pasado para tener una idea del ambiente en el que se está desarrollando, no hay ninguna estadística representada con gráfica, pero en la terminal se puede ver la cantidad de pasos que ha dado el agente, la vida actual del agente y cuantas basuras ha recolectado, se espera para un futuro, mejorar este programa para que pueda existir la visualización en la pantalla principal que verá el usuario.

A screenshot of a terminal window with a dark background and light-colored text. It displays the state of an agent named 'Agent 1000' at five different points in time. The text shows the agent's current steps, life value, and actions taken, such as getting a step or removing trash. The data is as follows:

Agent	Steps	Life	Action
Agent 1000	73	27	Initial state
Agent 1000	74	26	get_step:80
Agent 1000	74	26	get_step:81
Agent 1000	-	-	Removed trash 33 at (5, 6)
Agent 1000	74	26	Final state

Imagen 1: Fragmento de la terminal donde se muestran algunos valores que se recolectan en la simulación.

Conclusiones.

- El modelo de agentes propuesto logra simular un entorno dinámico donde los agentes toman decisiones para recolectar basura y gestionar su energía.
- La implementación de la recarga de energía en el punto de carga permite a los agentes prolongar su vida y continuar sus tareas (No se en que momento se rompió el código ya que al ponerle 100 de vida directamente funcionó la primera ejecución y después ya no funcionó más).
- El diseño modular y la arquitectura basada en eventos facilitan la extensibilidad y la modificación del modelo para abordar problemas más complejos o introducir nuevos comportamientos como será en el reto de este bloque.

- Dormir un poco y entregar la tarea un día tarde fue vital para poder atender esta problemática con más calma y mejores resultados, sin ser los óptimos, pero se logró algo.