



Tecnológico de Monterrey

Campus Santa Fe

MA. Actividad: Roomba

Modelación de Sistemas Multiagentes con Gráficas Computacionales

TC2008B.302

Profesor: Octavio Navarro

Fernanda Cantú Ortega A01782232

16 de noviembre de 2023

Problema que se está resolviendo y la propuesta de solución

El problema abordado se centra en la simulación de un entorno de limpieza inspirado en la aspiradora inteligente, Roomba. El objetivo es diseñar un modelo donde un agente pueda percibir su entorno y tomar decisiones efectivas para limpiar las celdas sucias, evitar los obstáculos, y gestionar sus niveles de batería con ayuda de estaciones de carga. La propuesta de solución es una simulación basada en Agent-Based Modelling con ayuda de la librería Mesa de Python.

Diseño de los agentes

Para la solución del problema, se crearon cuatro tipos de agentes distintos: el agente limpiador, los obstáculos, las celdas sucias, y las estaciones de carga. El objetivo del agente limpiador es recorrer la parrilla del modelo con el fin de limpiar todas las celdas sucias, siempre evitando los obstáculos. El agente limpiador cuenta con una batería limitada, por lo cuál es esencial que el agente reconozca cuando llega al 20% de su batería y que busque el camino de regreso a su estación de carga.

Con respecto a la capacidad efectora, el agente limpiador cuenta con un breve repertorio de acciones a su disposición. Para esta simulación, el agente limpiador puede moverse siguiendo la vecindad de Moore, es decir que puede moverse arriba, abajo, a la derecha, a la izquierda, y a todas sus diagonales, siempre tomando en cuenta el centro. El agente es proactivo, pues exhibe un comportamiento dirigido a cumplir su meta de limpiar todas las celdas. Esto se debe a que está programado para analizar las celdas que lo rodean y priorizar el movimiento hacia la primera celda sucia que encuentre. También, el agente tiene la capacidad de modificar su entorno pues cada vez que su posición coincide con la de una celda sucia, la celda sucia se “limpia”.

Para calcular las métricas de desempeño del agente limpiador, se agregaron al modelo distintas gráficas. Las métricas a medir son el tiempo necesario para que todas las celdas estén limpias, o se haya llegado al tiempo máximo (definido al inicializar la simulación), el porcentaje de celdas limpias al término de la simulación, y el número de movimientos realizados por el agente.

Arquitectura de subsunción

La arquitectura de subsunción de este modelo consiste en 4 simples acciones de parte del agente: moverse entre celdas, buscar las celdas sucias y limpiarlas, regresar a la estación de carga en caso de tener batería baja, y evitar los obstáculos. Las cuatro acciones son importantes, pues dependen una de la otra. Por ejemplo, para buscar las celdas sucias, el agente necesita moverse. Sin embargo, para moverse necesita tener batería. Para regresar a su estación de carga, necesita evitar los obstáculos. Se podría decir que es una arquitectura de Reflex Agents with State, pues el agente limpiador tiene como objetivo limpiar toda la rejilla y para lograrlo necesita analizar el mundo, ver cómo sus acciones afectan al mundo, decidir qué acción tomar, y repetir hasta cumplir su meta.

Características del ambiente

El ambiente consiste en una rejilla de 15x15 que cuenta con un borde de obstáculos, y con varios obstáculos posicionados aleatoriamente en las celdas restantes. El ambiente es accesible, puesto que el agente puede obtener información completa y precisa sobre el estado de este. También se programó un ambiente determinista, pues no hay incertidumbre con respecto a los efectos de las acciones que lleva a cabo el agente. Otro aspecto importante del ambiente es el hecho de que el agente puede tomar decisiones observando solamente el episodio actual, y no necesita pensar en futuros episodios. Se utilizaron sliders para que el usuario tenga control sobre el porcentaje de la rejilla que quiere llenar de obstáculos. Por defecto, el 65% de la rejilla tiene obstáculos. Sucede exactamente lo mismo con las celdas sucias: se posicionan aleatoriamente y el usuario puede modificar el porcentaje con un slider. Esto logra que el ambiente sea dinámico, pues el agente no tiene control sobre la cantidad de obstáculos ni de celdas sucias.

Estadísticas recolectadas en las simulaciones

Durante las simulaciones, se recopilan estadísticas para evaluar el desempeño del sistema y de los agentes. Entre estas se incluyen el tiempo necesario para limpiar todas las celdas, el porcentaje de celdas limpias al final de la simulación, y el número de movimientos realizados por el agente.

Conclusiones

Para concluir, este modelo ofrece una solución robusta, adaptable y escalable para simular eficientemente un ambiente donde un agente limpiador recorra celdas y las limpie. Gracias a las bases y fundamentos adquiridos tras la solución de esta problemática, se ha abierto un mundo de posibilidades que se pueden explorar utilizando agent-based modelling y simulaciones de sistemas.