Tecnológico de Monterrey

Campus Santa Fe

Nombre del trabajo:

Actividad: Roomba

Curso:

Modelación de sistemas multiagentes con gráficas computacionales

Estudiante:

Rebeca Davila Araiza

Matrícula:

A01029805

Docente:

Octavio Navarro Hinojosa

Gilberto Echeverría Furió

Fecha de entrega:

18 de noviembre

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

La problemática a resolver con el siguiente ejercicio es buscar una manera efectiva de movimiento para que el agente roomba pueda explorar todo el grid y limpiar las celdas sucias en el menor tiempo posible y sin que se le acabe la batería. Es por eso que la propuesta a esta solución es que el agente pueda recorrer el grid tomando las celdas que no ha visitado, y que ha visitado pocas veces en el grid hasta encontrar suciedad para limpiarla. Y que cuando tenga poca batería, encuentre el camino más corto, utilizando BFS (Breadth-First Search), hacia su respectiva celda de carga o la más cercana a este antes de que se le acabe la energía.

Objetivo General del Agente (Roomba/Agente Explorador)

El agente tiene como objetivo mantener el ambiente limpio buscando celdas con suciedad, desplazarse eficientemente para explorarlo todo y administrar su energía para evitar quedar inactivo. Además, debe regresar oportunamente a estaciones de carga para evitar quedarse sin batería.

Características de agentes

Capacidades Efectoras

- Los roomba eliminan los agentes de suciedad del grid cuando están encima de ellos.
- Si un roomba está ocupando una celda de carga, otro roomba no podrá usarla y tendrá que buscar otra celda de carga cercana

Percepción

El roomba puede percibir si:

- la celda actual contiene suciedad.
- la celda actual es una estación de carga.

- hay vecinos que son obstáculos.
- hay suciedad en celdas vecinas.
- hay celdas ya visitadas.
- Nivel de carga de batería propia.

Proactividad

El roomba actúa con base a las siguientes metas:

- Cuando detecta suciedad, prioriza limpiarla.
- Cuando su batería cae por debajo del umbral definido, sabe que debe regresar a un cargador.
- Si no hay suciedad visible, activa explora celdas no visitadas.
- Si ya visitó todas las celdas a su alrededor,

Reactividad

El roomba reacciona a los siguientes elementos del entorno:

- Evita obstáculos si los encuentra como vecinos.
- Se detiene a cargar cuando detecta una estación de carga y tiene batería baja.
- Si percibe suciedad cercana, interrumpe la exploración y cambia a modo “limpieza”.

Métricas de Desempeño

Solamente para un agente, ya que implementar estas métricas a mas de uno vuelve la simulación más lenta:

- Total de celdas limpiadas
- Energía gastada y regenerada por las estaciones de carga

Arquitectura de Subsunción

- Capa 3 – Exploración
    - El roomba se moverá a celdas no visitadas para seguir buscando celdas con suciedad
    - En cuanto detecta una celda vecina con suciedad, se moverá directamente a esa celda
- Capa 2 – Cargar bateria
    - El roomba cargará su bateria un 5% cada paso siempre y cuando se encuentre encima de una celda de carga y su batería no sea igual o mayor a 100%
- Capa 1 – Ir a la celda de carga más cercana
    - El roomba debe de mantener su batería mayor de 0% o no se moverá y no podrá cumplir el resto de tareas.
    - Si la batería llega a un límite bajo, su mayor prioridad será buscar el camino más corto a la estación de carga más cercana.
- Capa 0 – Limpieza
    - El roomba debe limpiar la suciedad en la celda donde se encuentra en cima actualmente

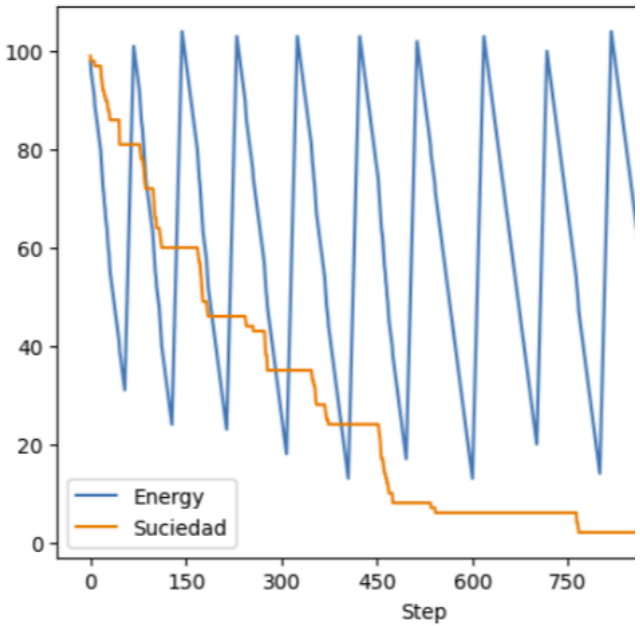
Características del Ambiente

- Es accesible ya que el roomba puede percibir el estado de su celda y vecinos, lo que le da información suficiente para tomar decisiones.
- Es determinista ya que las acciones tienen resultados predecibles. Por ejemplo: moverse a una celda siempre lleva al resultado esperado a menos que haya un obstáculo.

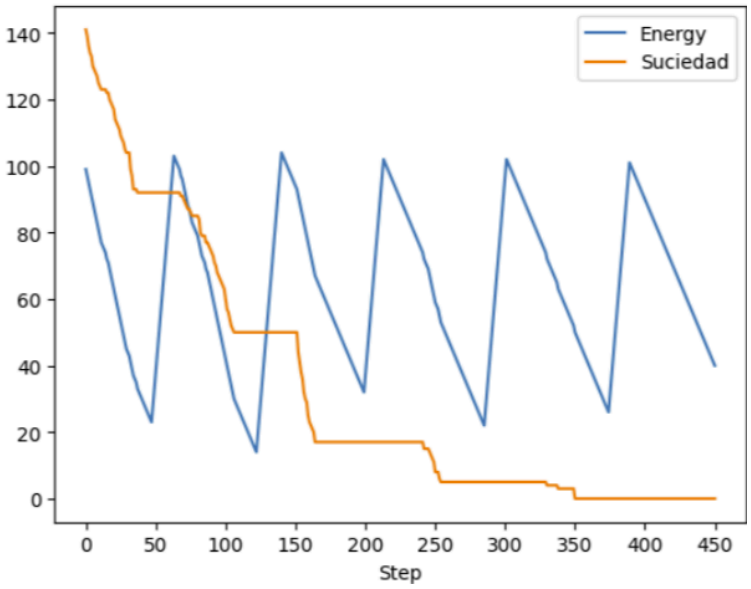
- Es secuencial ya que las decisiones actuales afectan estados futuros ya que pueden gastar la batería del roomba. y si llega a tener menos batería, sus movimientos y acciones cambian
- Es estático porqué el ambiente no cambia por sí solo, solo cambia con acciones del agente.
- Es discreto ya que se compone de celdas individuales con estados bien definidos.

Las estadísticas recolectadas en las simulaciones.

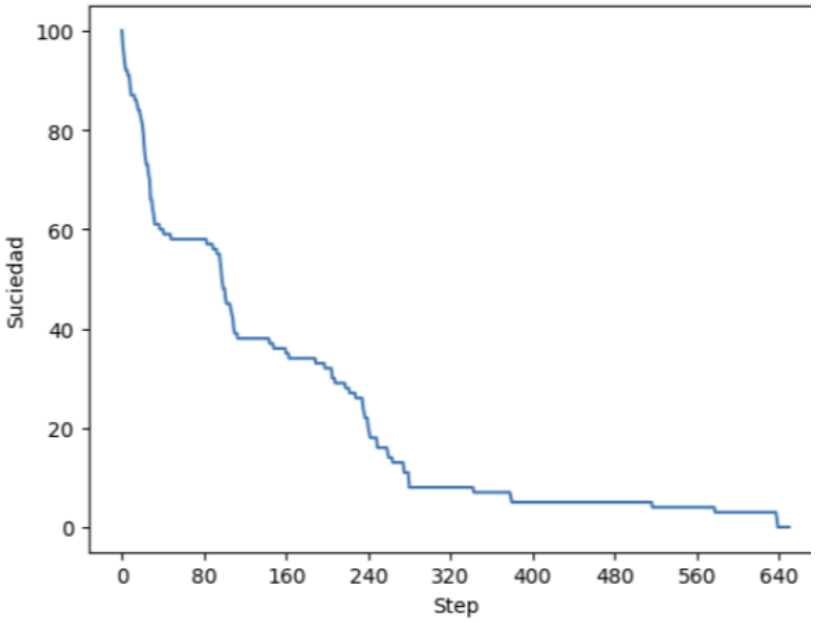
Grafica de un solo agente con grid de 20X20



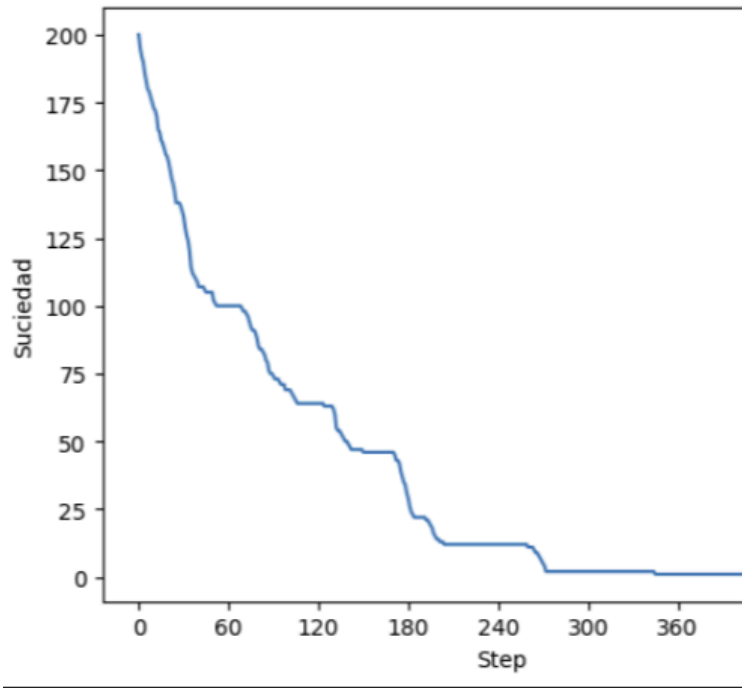
Grafica de un solo agente con grid de 15X15 con 35 obstáculos



Grafica de 4 agentes con grid de 28X28



Grafica de 4 agentes con grid de 28X28, pero con 200 de suciedad y 47 de obstáculos



Conclusiones.

El proceso para darle reactividad e inteligencia a los robots roomba fue bastante complejo. De hecho aún hay mucho que mejorar si se observan las gráficas. En las gráficas de solo un solo agente se pueden observar varias ocasiones donde el porcentaje de suciedad se mantiene estático. Esto podría ser por tres razones. Una puede ser ya que el agente se encuentra cargando y no realiza movimientos para limpiar. La segunda puede ser el camino que el agente toma de regreso a su última posición, la cual no tiene suciedad y no está limpiando nada. Y la tercera es que al llegar a su última posición, cabe la posibilidad de que vuelva a un área que ya esté completamente limpia y debe tomarse su tiempo para explorar el grid hasta que por fin encuentre suciedad. Esto también ocurre cuando hay más de un agente en el grid. La limpieza es mucho más rápida al tener más roombas limpiando, pero todavía

cabe la posibilidad de que exploren un área ya limpia, sobre todo cuando ya quedan pocas celdas sucias.

Es por eso que yo recomiendo la siguiente solución: dar un mejor algoritmo de movimiento a los roomba. Todavía recordando las celdas visitadas, pero teniendo alguna condición que le diga que ya está recorriendo el mismo lugar después de cierto número de pasos. Además que la selección de celdas no visitadas podría ser mucho más específica en lugar de seleccionar una random. Ya que eso arruina la dirección en la que el agente debe de ir para no ir a lugares ya limpios.