



Tecnológico de Monterrey

Campus Santa Fe

Integración de seguridad informática en redes y sistemas de software

Actividad: Rumba

Octavio

Presentado por:

Lucía Barrenechea A01782021

16 de noviembre del 2023

Introducción

El objetivo de este proyecto consistió en desarrollar un programa en Python mediante la implementación de la biblioteca Mesa, con el fin de simular el desempeño de la herramienta de limpieza "Rumba". Se diseñaron dos simulaciones: la primera con un solo agente y la segunda con múltiples agentes. Para asegurar el correcto funcionamiento de la simulación, se definieron tanto el ambiente como los agentes que participarán en ella. Se establecieron prioridades específicas para el agente Rumba, orientadas a que su comportamiento pueda reaccionar de manera efectiva ante las condiciones del ambiente. Este enfoque proporciona un marco sólido para explorar y comprender el comportamiento del agente de limpieza en entornos simulados, permitiendo así análisis detallados y la evaluación de su eficacia en diferentes contextos.

Desarrollo

Simulación 1

Estado Actual y Trigger	Condición	Acción
Rumba realiza una acción (step)	Si la simulación sigue corriendo.	La batería del Rumba disminuye -1.
Rumba detecta una basura en sus vecinos.	Si Rumba tiene más batería que 60.	Rumba se moverá al espacio de la basura.
Rumba se encuentra en la posición de una gente de basura	Si Rumba tiene suficiente batería	Rumba limpia esa basura.
Rumba detecta un espacio vacío y no visitado.	Si Rumba tiene suficiente batería y no detecta basura en otro espacio.	Rumba se mueve hacia ese espacio.
Rumba detecta un espacio vacío	Si Rumba tiene suficiente batería, no detecta basura en otro espacio y no detecta un lugar vacío que no ha sido visitado.	Rumba se mueve hacia el lugar vacío
Rumba tiene batería menor a 60.	Si Rumba tiene suficiente batería.	Rumba empieza a moverse a su estación de carga utilizando sus posiciones conocidas.
Rumba llega a su estación de carga.	Si Rumba necesita recargarse.	Rumba frena en la estación de carga y su batería regresa a 100
Rumba sale de la estación de carga	Si Rumba tiene suficiente batería.	Rumba regresa a su última posición

Si se inicia la simulación	Si se conecta a la interfaz,	Se coloca el borde en el grid.
Si se inicia la simulación	Si el número de obstáculos es mayor a 0	Se colocan los obstáculos de forma aleatoria en el tablero.
Si se inicia la simulación	Si existen agentes y existen celdas vacías.	Se añade el agente al grid.
Si se añaden agentes a la simulación	Si se guardan las posiciones iniciales de esos agentes correctamente	Se añade la estación de carga al punto (1,1)
Si todos los agentes de basura han sido borrados	Si el número de agentes de basura borrados es el mismo que el número de agentes de basura creados.	La simulación se para.

Simulación 2

Estado Actual y Trigger	Condición	Acción
Rumba detecta una basura en sus vecinos.	Si Rumba tiene más batería que 60 y no existe otra Rumba en ese espacio.	Rumba se moverá al espacio de la basura.
Rumba se encuentra en la posición de una gente de basura	Si Rumba tiene suficiente batería	Rumba limpia esa basura.
Rumba detecta un espacio vacío y no visitado.	Si Rumba tiene suficiente batería, no detecta basura en otro espacio y no existe otra Rumba en ese espacio.	Rumba se mueve hacia ese espacio.
Rumba detecta un espacio vacío	Si Rumba tiene suficiente batería, no detecta basura en otro espacio, no detecta otra Rumba en ese espacio y no detecta un lugar vacío que no ha sido visitado.	Rumba se mueve hacia el lugar vacío
Rumba detecta una estación de carga	Si rumba tiene batería mayor a 60	Rumba guarda la posición de esa estación en la lista de estaciones.
Rumba tiene batería menor a 60.	Si Rumba tiene suficiente batería.	Rumba empieza a moverse a su estación de carga más cercana utilizando sus posiciones conocidas.
Rumba llega a su estación de carga.	Si Rumba necesita recargarse y no existe otra	Rumba frena en la estación de carga y su batería

	rumba en esa estación.	regresa a 100
Rumba sale de la estación de carga	Si Rumba tiene suficiente batería y no hay otra rumba en ese espacio.	Rumba regresa a su última posición

Si se inicia la simulación	Si se conecta a la interfaz,	Se coloca el borde en el grid.
Si se inicia la simulación	Si el número de obstáculos es mayor a 0	Se colocan los obstáculos de forma aleatoria en el tablero.
Si se inicia la simulación	Si existen agentes y existen celdas vacías.	Se añaden los agentes al grid
Si se añaden agentes a la simulación	Si se guardan las posiciones iniciales de esos agentes correctamente	Se añaden las estaciones de carga.
Si todos los agentes de basura han sido borrados	Si el número de agentes de basura borrados es el mismo que el número de agentes de basura creados.	La simulación se para.

Análisis de Resultados

Tanto la simulación 1 como la simulación 2 exhiben un rendimiento adecuado al lograr la limpieza de todas las celdas en el entorno. Al realizar múltiples ejecuciones, se observó que, en el caso de la simulación 2, se completaba más rápidamente cuando las estaciones de carga estaban ubicadas a distancias considerables entre sí. Esta disposición permitía a los agentes explorar y cubrir un área más extensa de manera eficiente. Por otro lado, en la simulación 1, se evidenció una mayor eficacia cuando las basuras estaban concentradas en áreas específicas o distribuidas uniformemente en la cuadrícula. Además, se notó que el aumento en el número de agentes contribuía a acelerar la finalización de la simulación. Este hallazgo destaca la influencia de la densidad de obstáculos y la distribución de las estaciones de carga en el rendimiento y eficiencia de los agentes en ambientes multiagente.

Simulación 1

Current Step: 0

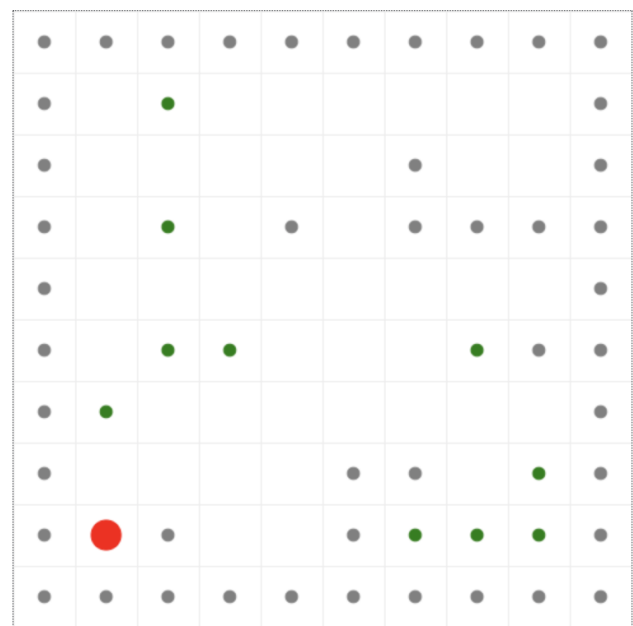


Imagen 1: Visualización del grid con todos sus componentes.

Current Step: 79

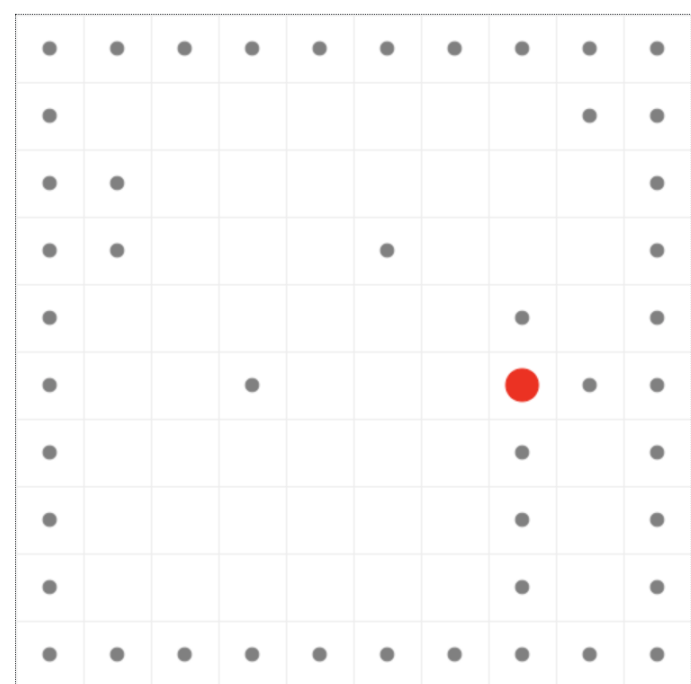
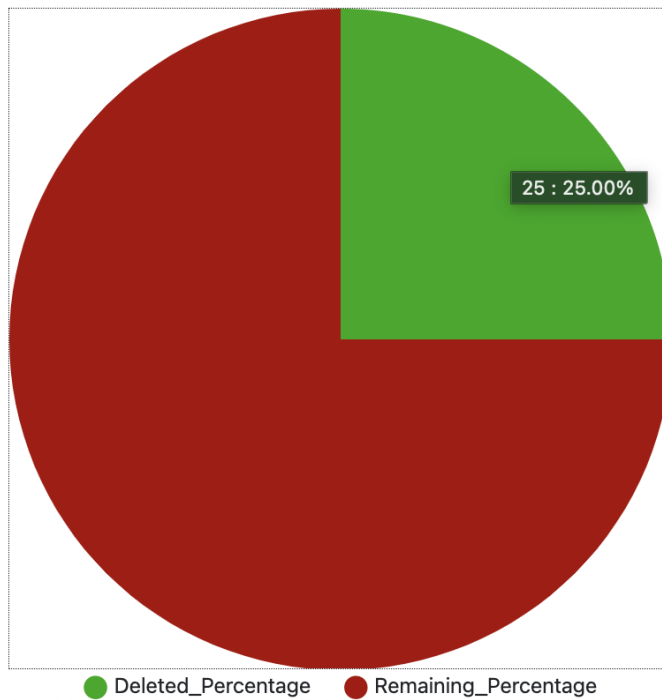
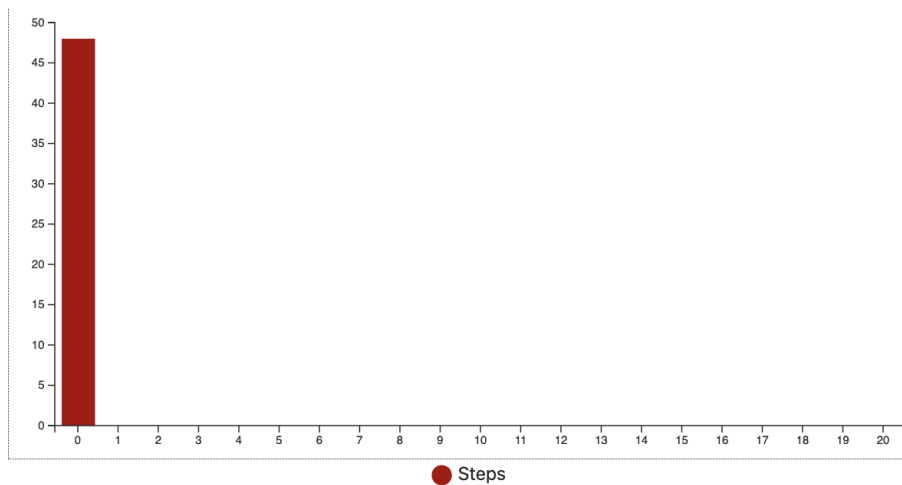


Imagen 2: Visualización de la simulación terminada



Gráfica 1: Comparación de basuras recogidas y basuras existentes



Gráfica 2: Cantidad de pasos realizados por el agente.

Simulación 2

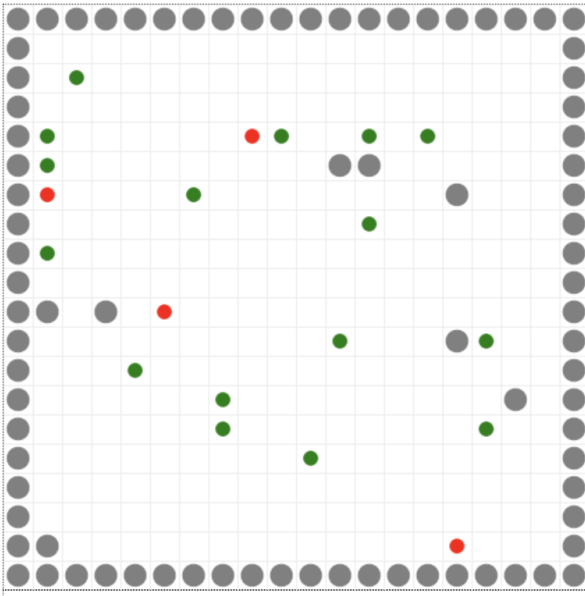


Imagen 3: Visualización del grid con todos sus componentes excepto las estaciones de carga.

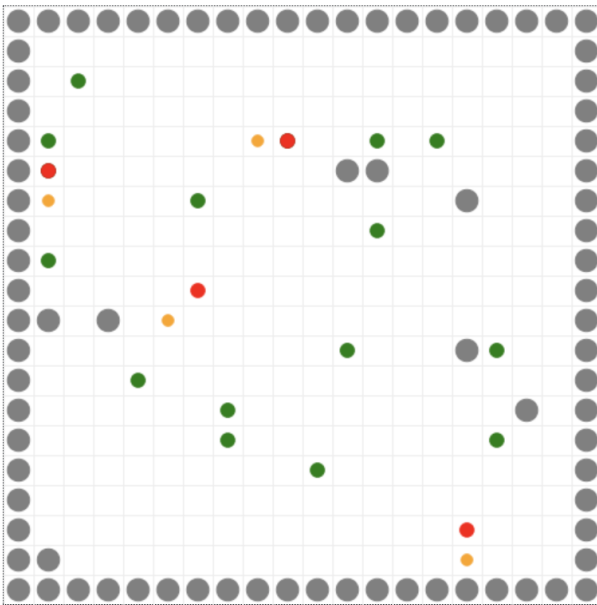
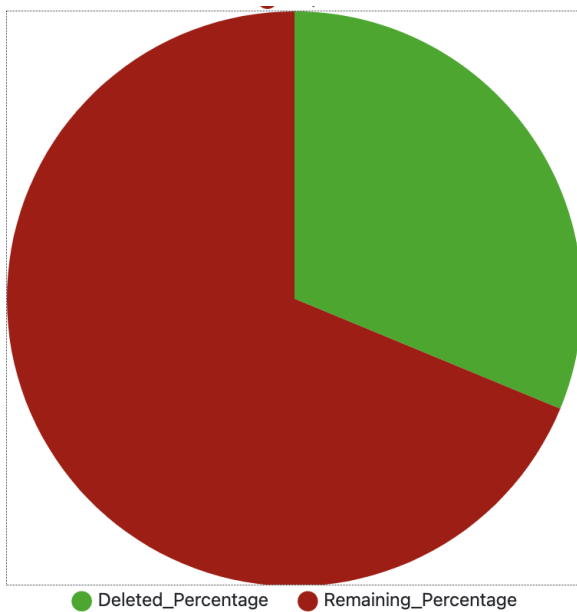
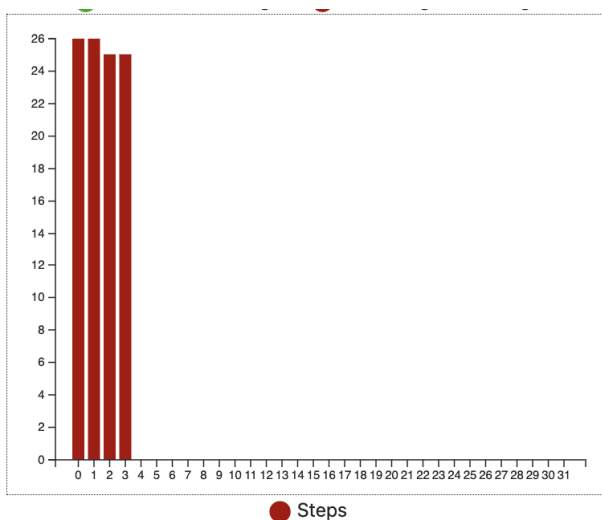


Imagen 4: Visualización del grid con todos sus componentes.

Imagen 5: Visualización de la simulación terminada



Gráfica 3: Comparación de basuras recogidas y basuras existentes



Gráfica 4: Histograma mostrando cada step de las Rumbas

Conclusiones

Ambas simulaciones fueron ejecutadas con éxito, logrando alcanzar sus objetivos de manera precisa. La implementación de estas técnicas revela la versatilidad y adaptabilidad de los modelos de agentes en diversos contextos. La incorporación de múltiples agentes añadió una capa de complejidad intrigante, observando cómo interactúan y reaccionan ante el entorno. Este enfoque multiagente demuestra ser prometedor, ofreciendo una perspectiva valiosa para abordar problemáticas complejas a través de la colaboración y la adaptación dinámica de los agentes en el sistema simulador.