

Reporte - Evidencia 1. Actividad Integradora

Equipo: NA

Grupo: 301

Integrantes:

Diego Vega Camacho - A01704492

Profesor:

Pedro Oscar Pérez Murueta

19 de Noviembre , 2023

Estrategia de desarrollo

La metodología que se siguió durante la implementación del código comienza conociendo lo que se nos solicita, en la definición del problema se nos pide la creación de un modelo de simulación de robots limpiadores que interactúan con cajas y torres en un entorno. Además, se especifica que la simulación tiene un objetivo específico relacionado con la marcación y llenado de torres.

Entonces, teniendo en cuenta el contexto del problema a resolver, se buscó seguir una metodología por pasos la cual se basaba en el siguiente orden (el código no sigue el mismo orden, esto solo fue la lógica de razonamiento):

1. Diseño del Modelo
2. Implementación de Agentes
3. Implementación del Modelo
4. Visualización y Recopilación de Datos
5. Análisis y Optimización

Durante la primera etapa se diseñó el modelo (BoxRobotModel) para representar el entorno y los agentes (robots y cajas). De igual forma se identificaron los atributos y comportamientos necesarios para los agentes.

Posteriormente en la implementación de agentes se buscó diseñar e implementar los agentes (CleanerRobot y Box) con los atributos y funciones específicas que buscaba cumplir el modelo. Además, se crearon funciones para la toma de decisiones de los robots, como buscar torres, buscar cajas, y moverse.

En la parte de la implementación del modelo, se implementó el modelo y en él se incluye la colocación aleatoria de robots y cajas en la cuadrícula.

Como penúltima etapa se implementaron funciones para recopilar datos durante la simulación, lo que facilitaría la visualización y el análisis posterior. Se utilizó matplotlib para generar visualizaciones de la simulación y analizar los resultados .

Finalmente, en la última etapa se buscaron oportunidades para simplificar la lógica y mejorar la eficiencia de las funciones una vez que se había acabado toda la implementación, esto con la finalidad de reducir los pasos y poder generar una implementación más eficiente.

Ahora explicaré de forma detallada las implementaciones de los agentes y el modelo

Agent:

- CleanerRobot: Representa a los robots limpiadores. Cada robot tiene atributos para gestionar si lleva una caja, si tiene una torre, las coordenadas de la torre y la caja que busca, entre otros.
- Box: Representa las cajas presentes en el entorno.

Model:

- BoxRobotModel:
 1. Gestiona la creación del entorno, la colocación de robots y cajas, y la ejecución de la simulación.
 2. Utiliza una rejilla para representar el entorno y un programador para gestionar la activación aleatoria de los agentes.
 3. Recoge datos durante la simulación para su posterior visualización.

Reflexión

Durante la realización de esta actividad pude integrar las diferentes formas de implementación de los diferentes sistemas de multiagentes que fuimos evaluando y revisando, con esto, pude ajustar las implementaciones de los mismos al nuevo sistema multiagente que se nos solicitaba.

¿Existe una mejor solución en cuanto a la cantidad de pasos que se requieren para completar la actividad?

Si, considero que al ir reduciendo la cantidad de for's que utiliza el problema para poder ejecutarse podríamos utilizar list comprehensions en lugar de bucles for para hacer el código más conciso y, en algunos casos, más eficiente. De igual forma podríamos eliminar bucles innecesarios, algunos bucles y condicionales redundantes se pueden eliminar o simplificar para reducir la complejidad y mejorar la eficiencia.

Finalmente considero que podría utilizar más funciones integradas de Python para mejorar la legibilidad y, en algunos casos, la eficiencia.