****



**Modelación de sistemas multiagentes con gráficas computacionales**

**Reporte de la Solución**

Miguel Angel Becerra Ayala A01710076

Santiago Yael Nieto A01612348

Santiago Palacios Menes A01276169

**Profesores:**

Pedro Oscar Pérez Murueta

Denisse Lizbeth Maldonado Flores

Alejandro Fernández Vilchis

**13 de junio de 2025**

**Liga del Repositorio de GitHub**

<https://github.com/Miguel-2004/Medieval_Rescue.git>

**Liga a Presentación**

[**Medieval Rescue.pptx**](https://docs.google.com/presentation/d/1OhL4G1aRwDrBlqFpiSvqIOQAVnz9tBKb/edit?usp=sharing&ouid=110574947673457559547&rtpof=true&sd=true)

**Analisis de la Solución**

Durante la implementación del sistema multiagente para la simulación del juego Flash Point Fire Rescue tuvimos diferentes desafíos ya que teníamos que crear un estrategia para la toma de decisiones, en especial para la optimización de rescate de víctimas y la gestión de obstáculos (paredes, puertas). A lo largo de este reporte analizaremos de manera exhaustiva los dos enfoques desarrollados para abordar el problema. Por una parte tenemos la estrategia aleatoria basada en la exploración no dirigida y una estrategia inteligente basada en algoritmos de búsqueda para rutas óptimas.

Por una parte tenemos la estrategia aleatoria que le dimos un enfoque de exploración donde se prioriza la cobertura del espacio sobre la eficiencia. Los agentes (Bomberos) tienen un sistema de movimiento pseudo aleatorio que tiene incorporado un sistema de memoria para evitar patrones repetitivos. Esta estrategia utiliza una estructura de datos circular que almacena las últimas posiciones visitadas, permitiendo que los agentes eviten regresar de manera inmediata a la misma posición. El algoritmo implementado evalúa direcciones válidas desde la posición actual y aplica un filtro que excluye las 3 últimas posiciones de la memoria. Cuando todas las direcciones válidas para el bombero están en la memoria, recurre a una selección de movimiento completamente aleatoria, para evitar el bloqueo del agente.

En cuanto a la gestión de objetivos, la estrategia aleatoria tiene un sistema de búsqueda muy limitado que solo identifica POIS y salidas 5 celdas alrededor. Tienen como prioridad romper puertas y paredes ya que no tienen una estrategia de movimiento que les ayude a hacer mejor las cosas, esto lo hacen de manera inconsciente.

Por otra parte tenemos la estrategia inteligente o definida y esta es como tener un equipo de bomberos experimentados ya con un plan hecho, Dijkstra nos ayuda en la definición de la ruta más eficiente para llegar a donde necesiten ir, nos ayuda en el ahorro de energía de los bomberos y además se planea que el daño al edificio sea menor. Dijkstra funciona como un mapa donde se muestra el costo de los movimientos y como se puede ahorrar energía y además ahorrar el daño que se le puede ocasionar al edificio.

Una de las cosas que cabe resaltar es que Dijkstra se actualiza dinámicamente y de esta manera se actualizan las paredes rotas o puertas abiertas, el fuego como esta propagado y esto ayuda en las decisiones de los demás bomberos debido a que es como si se estuviera informado en tiempo real de cuál es la situación, para la toma de decisiones.

Este tiene un enfoque de rescatar víctimas sobre la destrucción innecesaria de destruir el edificio. Los agentes calculan dinámicamente sus objetivos basándose en el estado, cuando no tienen una víctima estos navegan al Poi más cercano utilizando la distancia Manhattan como heurística de proximidad y lo hacen de igual manera para cuando tienen una víctima rescatada. El uso responsable de recursos energéticos garantiza el éxito de la misión.

En conclusión, el análisis demuestra claramente que el uso de los algoritmos adecuados y sofisticados sobre la estrategia aleatoria dan mejores resultados. La realización de esta actividad fue algo retador que nos llevó a pensar fuera de la caja lo que nos ayudó para el desarrollo de nuestras competencias.