

INIT.CO

FLASHPOINT SPACE RESCUE

INSPIRED BY
flashpoint fire rescue

Juan José Goyeneche - A01712547

Iñaki S. Pérez Lozada - A01278252

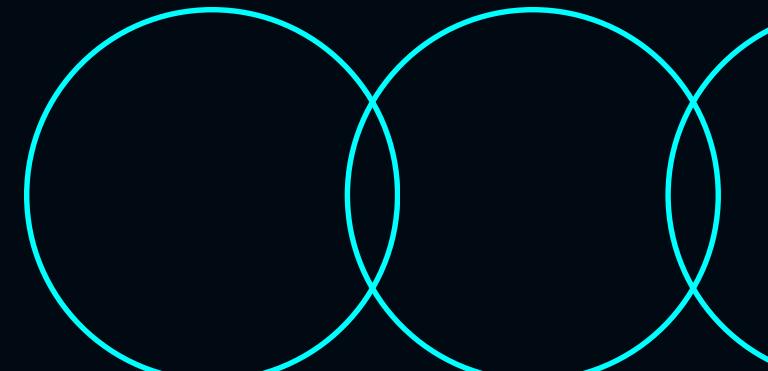
Luis Enrique Bartolo Macario - A01773034



Objetivo del proyecto

El objetivo del proyecto es demostrar cómo podemos utilizar un sistema multiagentes en conjunto con el uso de gráficas computacionales para resolver un problema complejo, en este caso ganar el juego de FlashPoint Space Rescue.

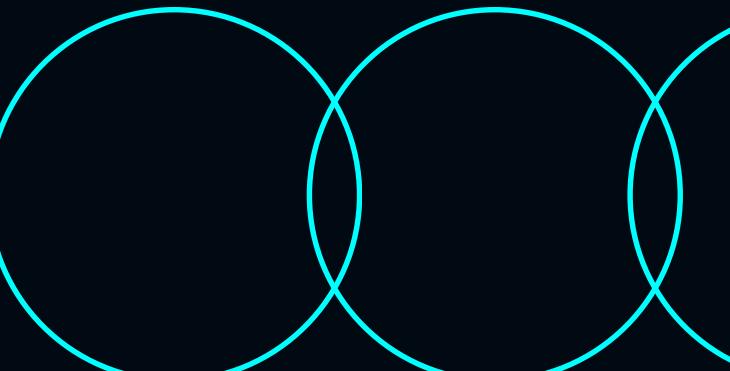
Usamos Unity para crear una simulación del funcionamiento de nuestro sistema multiagentes.



Nuestro mundo...



En los confines más oscuros de la galaxia, donde la luz de las estrellas apenas roza el vacío infinito, una señal de auxilio rompe el silencio: la nave Eclipsia ha quedado a la deriva, infestada por colosales arañas espaciales surgidas de una experimentación biológica. Ahora, mientras la nave se queda sin combustible un pequeño grupo de astronautas de rescate se embarca en una misión Peligrosa: abordar el casco moribundo, rescatar a los tripulantes atrapados y enfrentar a criaturas que acechan en cada corredor. En este borde olvidado del universo...

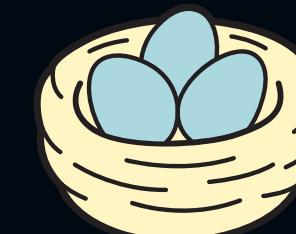




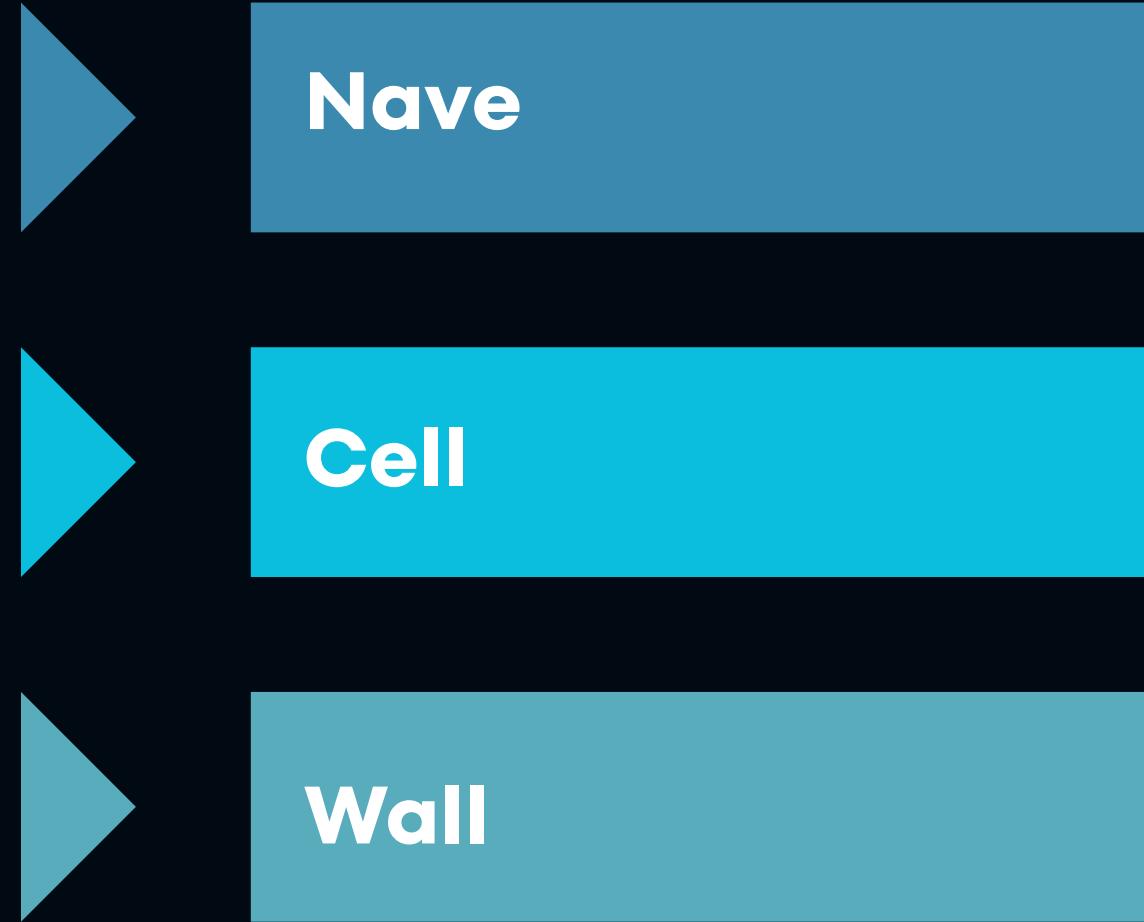
“

REGLAS Y OBJETIVO DEL JUEGO

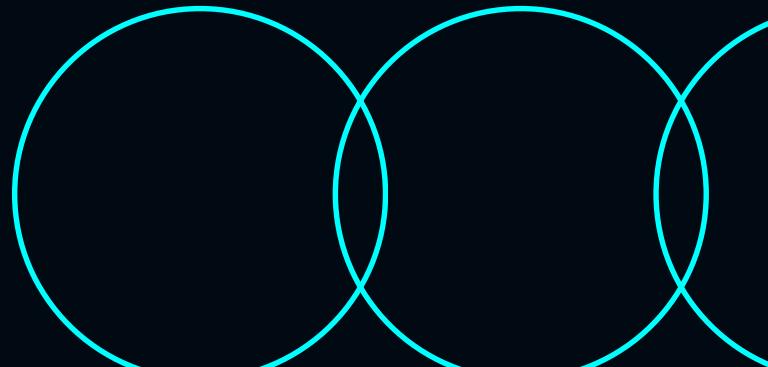
- Rescatar 7 víctimas antes de que el edificio colapse
- Moverse, abrir puertas, matar arañas o quitar huevos resta AP adicional
- Llegar a una salida restaura los AP's
- Cada knockdown quita 1AP, si llevas una víctima esta muere



Nave y ejecutar_turno()



Cada turno resetea los AP de los agentes y asigna tareas mediante subastas (en modo inteligente). Los agentes ejecutan su step() gastando AP para moverse, apagar fuego o rescatar víctimas. Tras cada acción, dados determinan dónde aparece humo o fuego, pudiendo causar explosiones que dañan el edificio. El turno finaliza verificando victoria (7 rescatadas) o derrota (4 perdidas / 24 daño).



Estrategia inteligente



ConVictima y EnSalida

➤ Rescatar Victima ()

Con Victima

➤ Ir a Salida ()

En POI Revelado

➤ Cargar Victima ()

En POI

➤ RevelarPOI()

Fuego Cell/ Cell Vecina

➤ Apagar Fuego ()

Hacer Tarea Asignada

Pos == Target

A*(Target)

HayFuego?

Mover

Priorización

r/boardgames • 8y ago
missedtrigger Magic The Gathering

Flash Point: Fire Rescue first time, 2p setup recommendations?

I have **Flash Point: Fire Rescue** on the way, and my first game will be 2 players. I'm a moderately experienced board gamer, and my wife is very sharp but rarely plays board games.

What does everyone recommend for first-time 2p setup? I've read some recommendations that each player control 2 firefighters... or that the 2 players share control of a 3rd firefighter manning the deck gun, for example. (Between those 2 options, I think the latter would be more manageable...)

I assume Experienced: Recruit level on the easier board (4 entrances) would be a good place to start? Any setup modifications?

r/boardgames • 10y ago
Gootch03

Flash Point: Fire Rescue, difficulty question

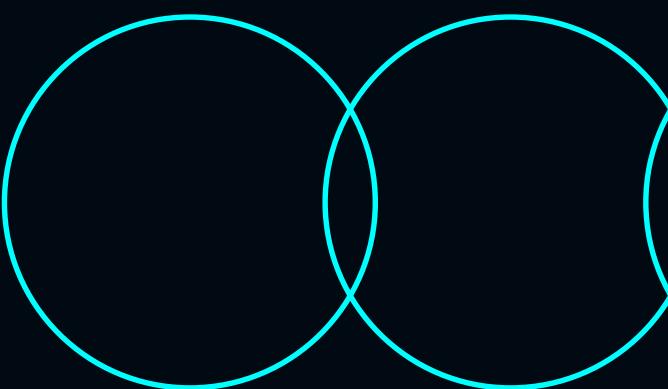
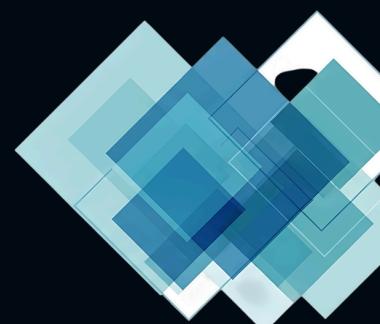
I played Flash Point for the first time with my group a few days back. We played the beginner version where several pieces and harder rules aren't used. We played a demo copy at a game store and didn't quite have the time to read the remaining instructions to see how it would affect later play throughs.

We really enjoyed the beginner game, gameplay was very quick, but it seemed to be very easy. Before I pick this up, do the other rules and later game plays provide more of a challenge? Also does the fun factor stay the same if we really enjoyed the basic game?

Our group likes co-op games and this seems like an easy enough one where everyone can do their own thing and not suffer from quarterbacking.

Edit

Lots of good stuff here. Any advice on what some of the better expansions are?



Hacer Tarea Asignada

Pos == Target

A*(Target)

HayFuego?

Mover

Class Subasta



Subasta Inversa (menorCosto)

- "llevar_victima"
- "rescatar_victima"
- "investigar_poi" (Max = x)
- "apagar_fuego"

- Lista de Agentes Sin Tarea y Sin cargar Victima.
- Lista de X targes (POIs, Fuego, Victimas).

```
for i in targets:  
    mejor_oferta = 9999  
    mejor_agente = None  
    for agente in Agentes:  
        oferta = agente.calcular_costo(i)  
        if oferta < mejor_oferta:  
            mejor_oferta = oferta  
            mejor_agente = a  
    asigna tarea  
    asigna target = i
```

Algoritmo A*

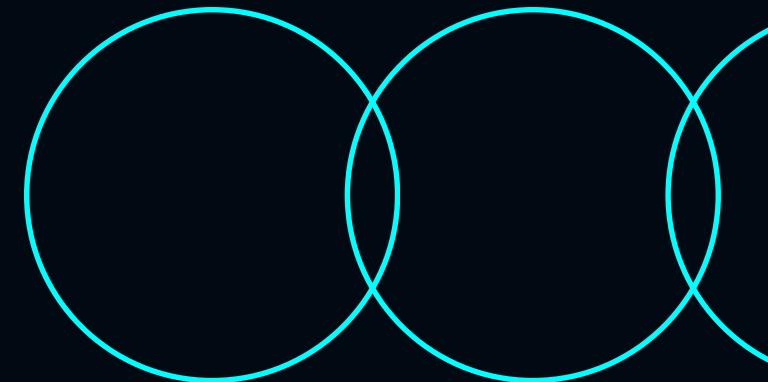
Descubriendo el camino...

¿Cuál es el camino más barato desde donde estoy hasta donde quiero llegar?

No buscamos el camino más corto, buscamos el menos costoso

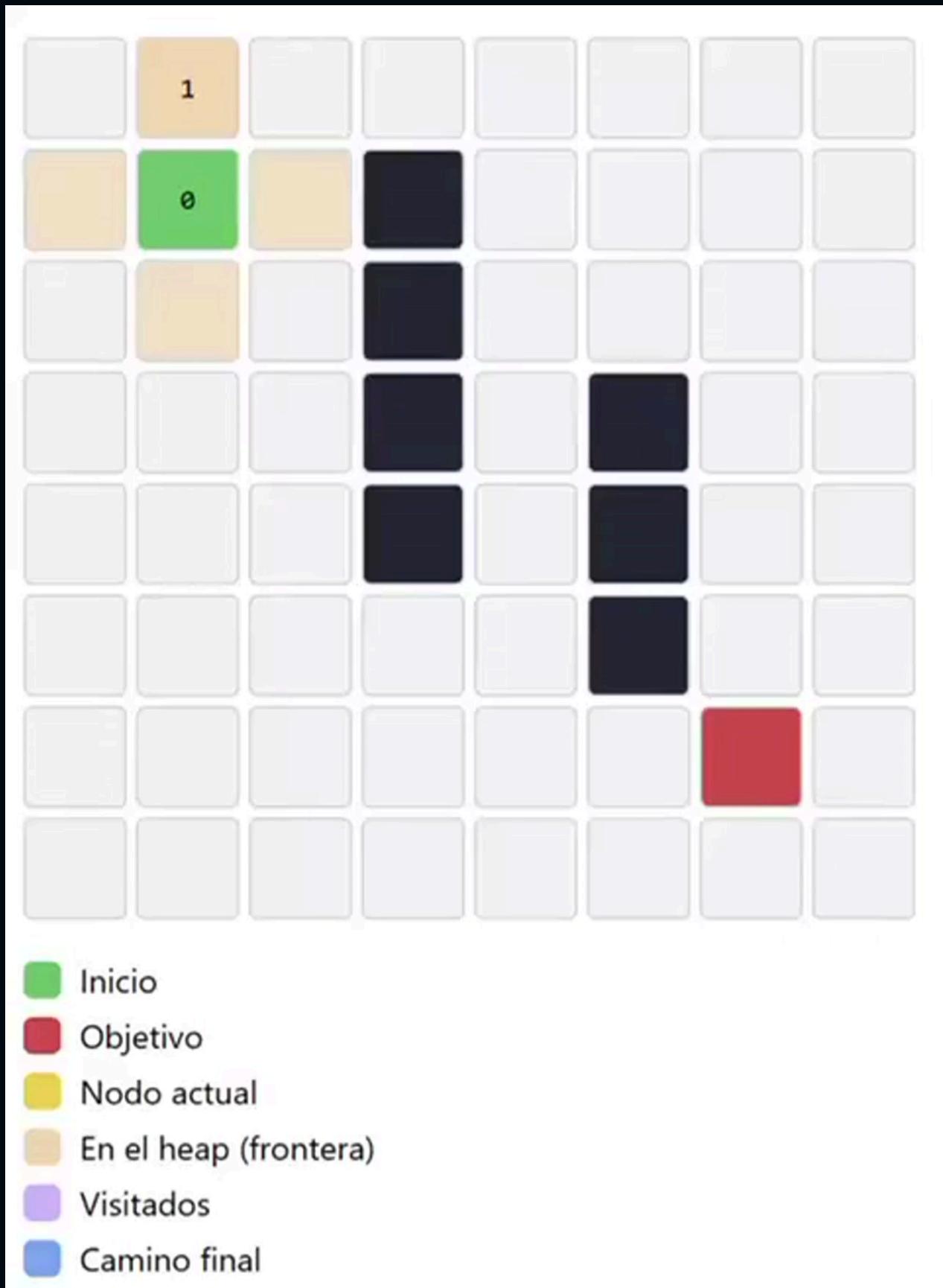
Los tres conceptos clave:

- g(costo real)
 - Si abrí 2 puertas y me moví 3 veces: $g = 2 + 3 = 5$ AP
- h(heurística - estimación)
 - ¿Cuánto creo que me falta para llegar al objetivo? Usamos distancia Manhattan $|x_1 - x_2| + |y_1 - y_2|$
 - Si estoy en (2, 3) y quiero llegar a (5, 7): $h = |2 - 5| + |3 - 7| = 3 + 4 = 7$
- f(costo total estimado)
 - $f = g + h$



Algoritmo A*

1. Agregamos el inicio a la frontera
2. Mientras la frontera NO ESTÉ VACÍA:
 - a. Saca el nodo con MENOR F
 - b. Si es la meta terminamos y reconstruimos el camino
 - c. Para cada vecino válido:
 - i. Calcula nuevo_g = g_actual + costo_del_paso
 - ii. Si es un camino mejor al que conocíamos:
 1. Actualiza g del vecino
 2. Calcula f = nuevo_g + h
 3. Agregamos a la frontera
3. Si la frontera se vacía NO HAY CAMINO



Resultados obtenidos

Corrimos la simulación 100 veces por cada estrategia obtuvimos los siguientes resultados:

--- Resumen General de 100 Simulaciones (Estrategia Inteligente) ---

Promedio de Víctimas Rescatadas por Simulación: 5.10

Promedio de Víctimas Perdidas por Simulación: 1.33

Promedio de Daño Estructural Final por Simulación: 21.52/24

Número de Victorias: 33 / 100

Número de Derrotas: 67 / 100

--- Métricas Promedio por Agente (a través de todas las simulaciones) ---

Fuegos Extinguidos (acumulado): 3.98

Humo Extinguido (acumulado): 5.79

POIs Revelados (acumulado): 1.65

Víctimas Transportadas (acumulado): 1.04

Knockdowns (final): 0.41

Porcentaje de Agentes Vivos al Final: 92.33%

--- Resumen General de 100 Simulaciones (Estrategia Random) ---

Promedio de Víctimas Rescatadas por Simulación: 0.34

Promedio de Víctimas Perdidas por Simulación: 1.05

Promedio de Daño Estructural Final por Simulación: 24.46/24

Número de Victorias: 0 / 100

Número de Derrotas: 100 / 100

--- Métricas Promedio por Agente (a través de todas las simulaciones) ---

Fuegos Extinguidos (acumulado): 2.66

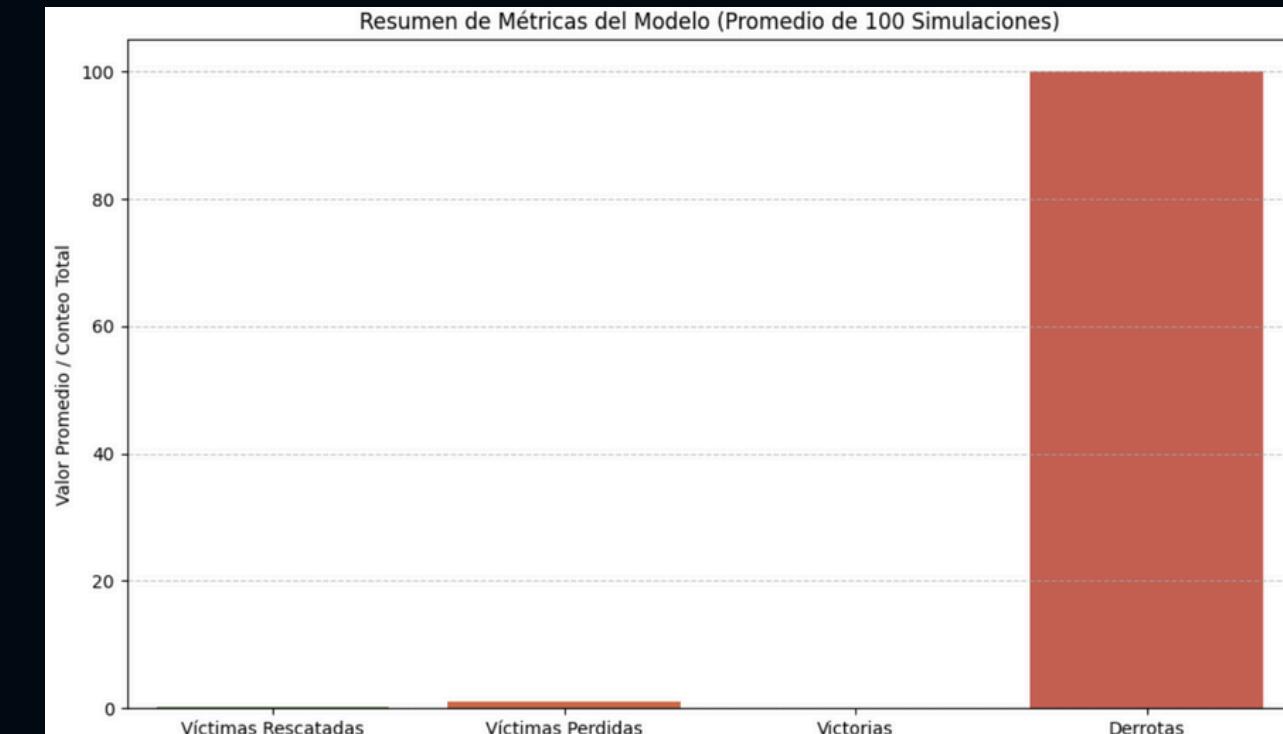
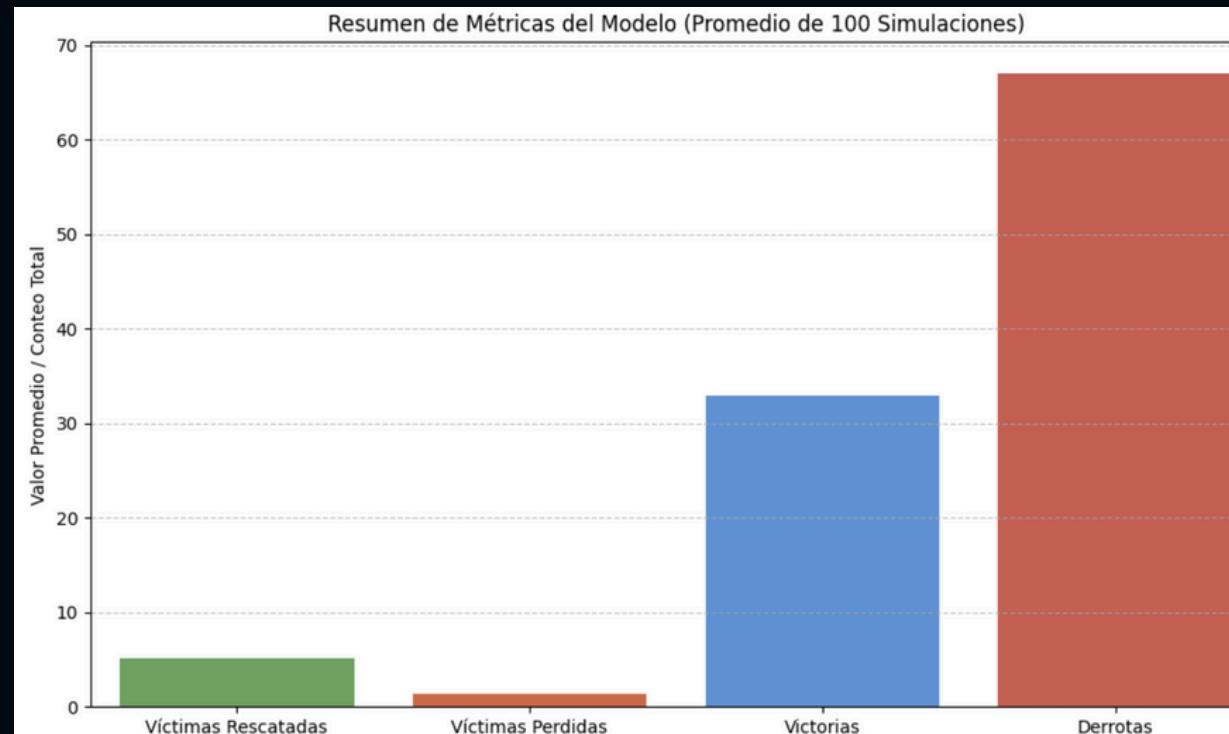
Humo Extinguido (acumulado): 0.00

POIs Revelados (acumulado): 0.31

Víctimas Transportadas (acumulado): 0.18

Knockdowns (final): 0.30

Porcentaje de Agentes Vivos al Final: 94.50%



Conexión Cliente-Servidor



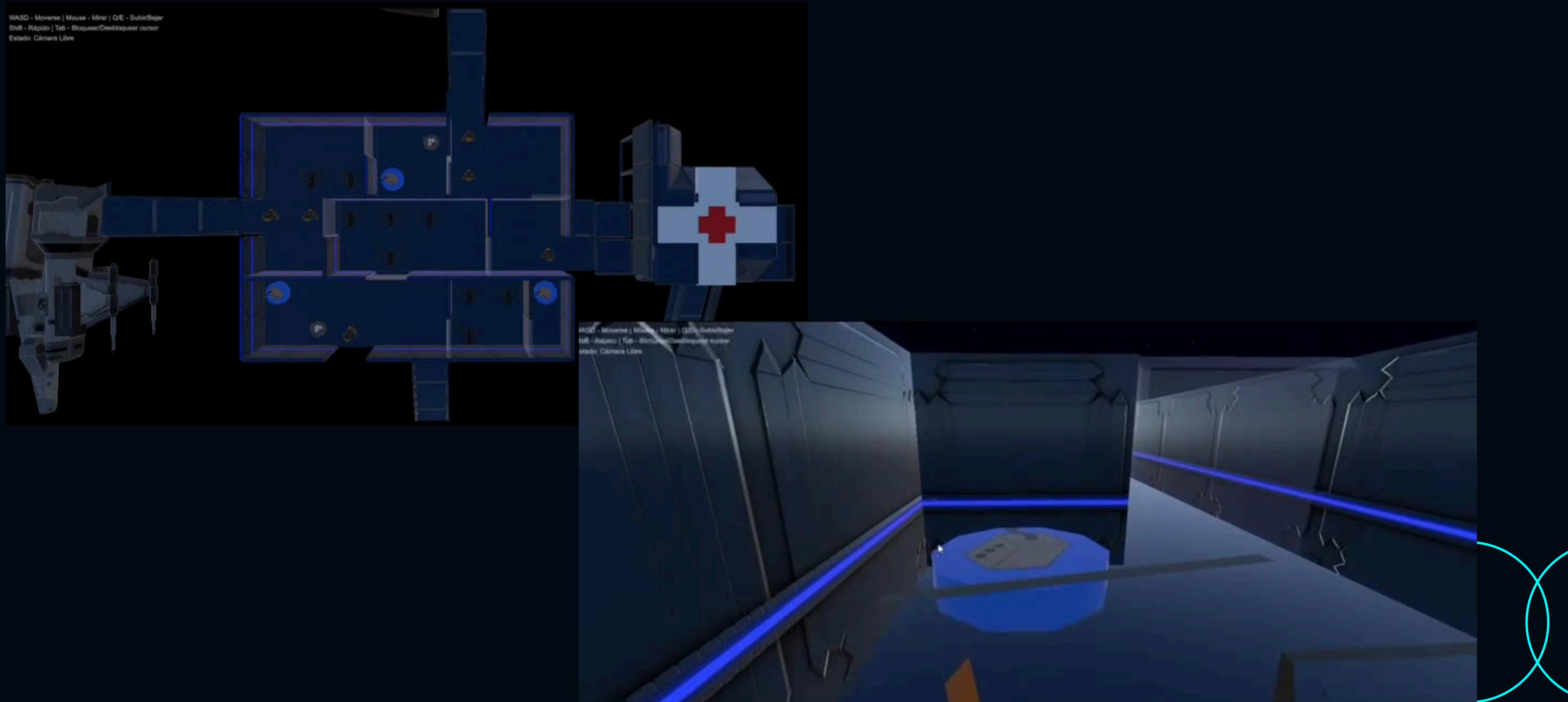
- Servidor Python: http.server en localhost:8585, endpoint GET /simulation_data.
- Genera simulacion.json
- Cliente Unity: UnityWebRequest consume el api
- TableroBuilder
- SimulacionRunner

Problemas

- Formato del json
- Hacer que el json construya correctamente el tablero
- Simulación en tiempo real o de un jalón



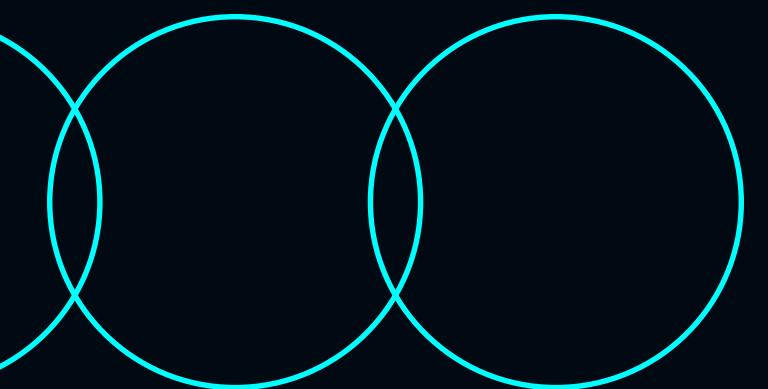
Simulación en Unity



Cómo podríamos mejorar?

En el sistema multiagente:

- Tomar en cuenta cierre de puertas para prevenir la expansión del fuego
- Hacer funciones para evitar código repetido
- Agregar tareas en subastas
- Mejorar asignación de tareas específicas



Conclusion

El sistema de subastas demostró que la asignación óptima supera al comportamiento aleatorio.



Uso de IA

Nos apoyamos de la IA para implementar soluciones a lo largo de nuestro proyecto, aunque todo el diseño y gran parte del sistema multiagentes fue desarrollado por nuestra cuenta, debido a nuestra falta de conocimiento técnico con respecto al uso de librerías como matplotlib, pandas o numpy gran parte del sistema de análisis y animaciones fueron desarrollados con el apoyo de esta herramienta, también la utilizamos como recurso para ideas o consulta sobre MESA y C#.



Uso de IA

Aquí tienes tu presentación sobre proyecto multiagentes generado con IA. 😊

LINK AL REPOSITORIO

LA EVIDENCIA SE ENCUENTRA EN LA RAMA
DEL RELEASE 01

[https://github.com/Luissitow/MODELACION
DEAGENTESTC2008B.301.git](https://github.com/Luissitow/MODELACIONDEAGENTESTC2008B.301.git)

Uso de IA

aaaaaaaaaa no cierto, lo hicimos nosotros,
piky promise.



GRACIAS

Estrategia y funcionamiento

Comprendiendo nuestra solución

- ¿Cómo funciona nuestro agente?
- Estrategia inteligente
- Algoritmo A*
- Distancia Manhattan
- Herramientas utilizadas

