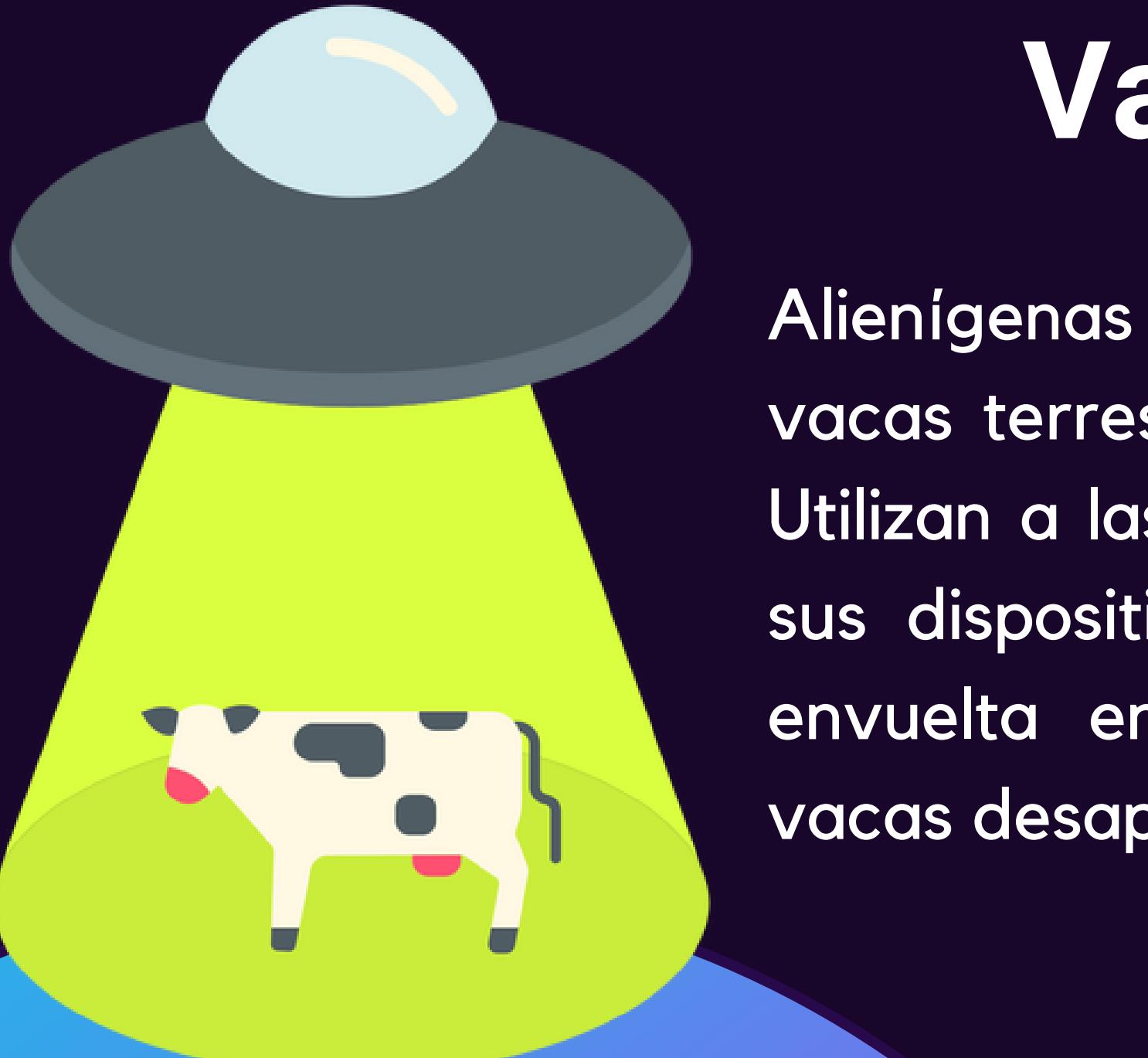




>>>



Alerta: Vacas Desaparecidas!

Alienígenas llamados "Lunaris" han descubierto que las vacas terrestres poseen una fuente de energía única. Utilizan a las vacas como catalizadores para alimentar sus dispositivos y naves espaciales. La Tierra se ve envuelta en un misterio intergaláctico mientras las vacas desaparecen hacia las estrellas.

MULTIAGENT FOOD COLLECTOR

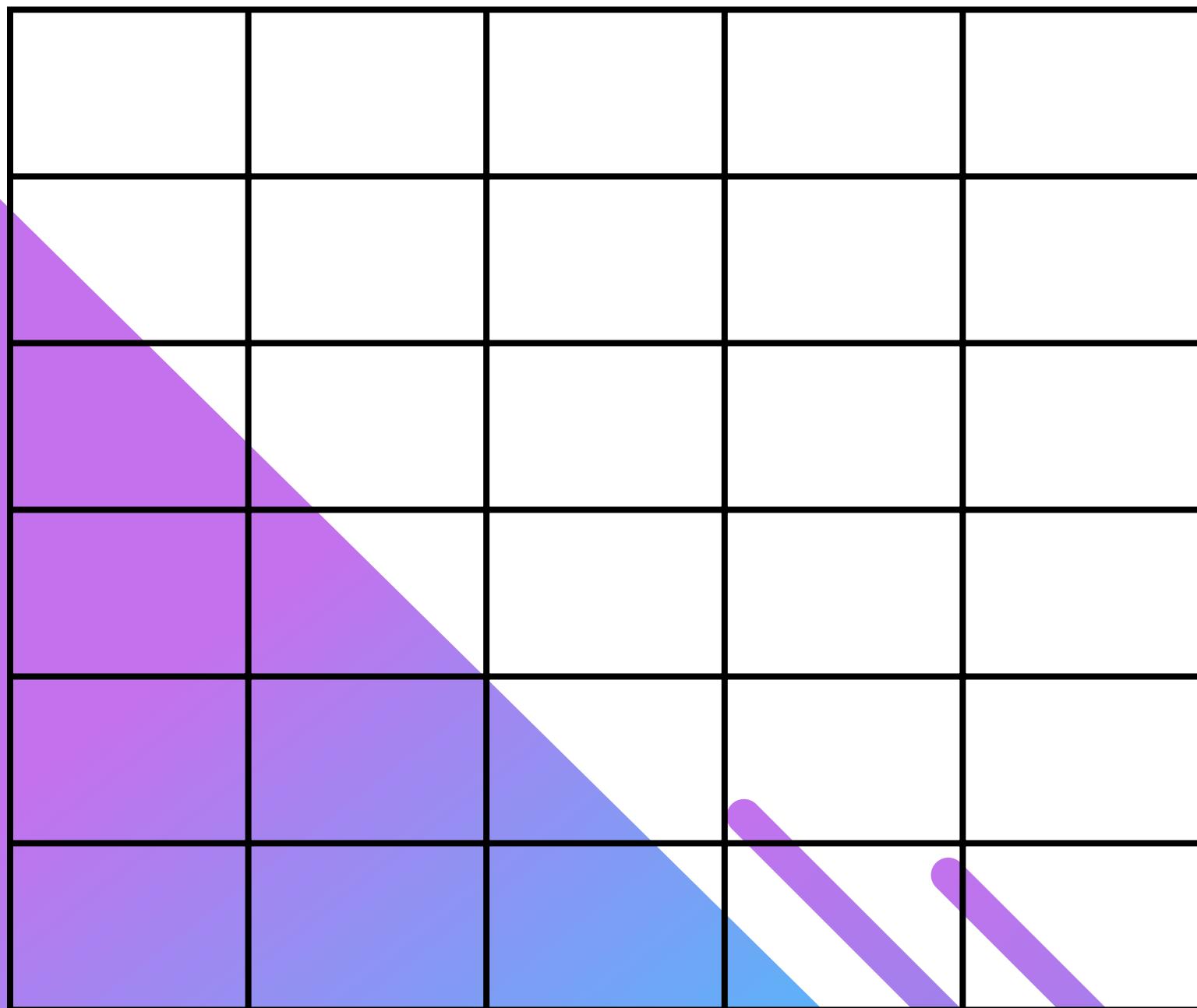
Presenta

DANIEL GUTIERREZ GOMEZ A01068056
DIEGO ERNESTO SANDOVAL VARGAS A01709113
JUAN MANUEL GONZALEZ ASCENCIO A00572003



Problema

20 x 20



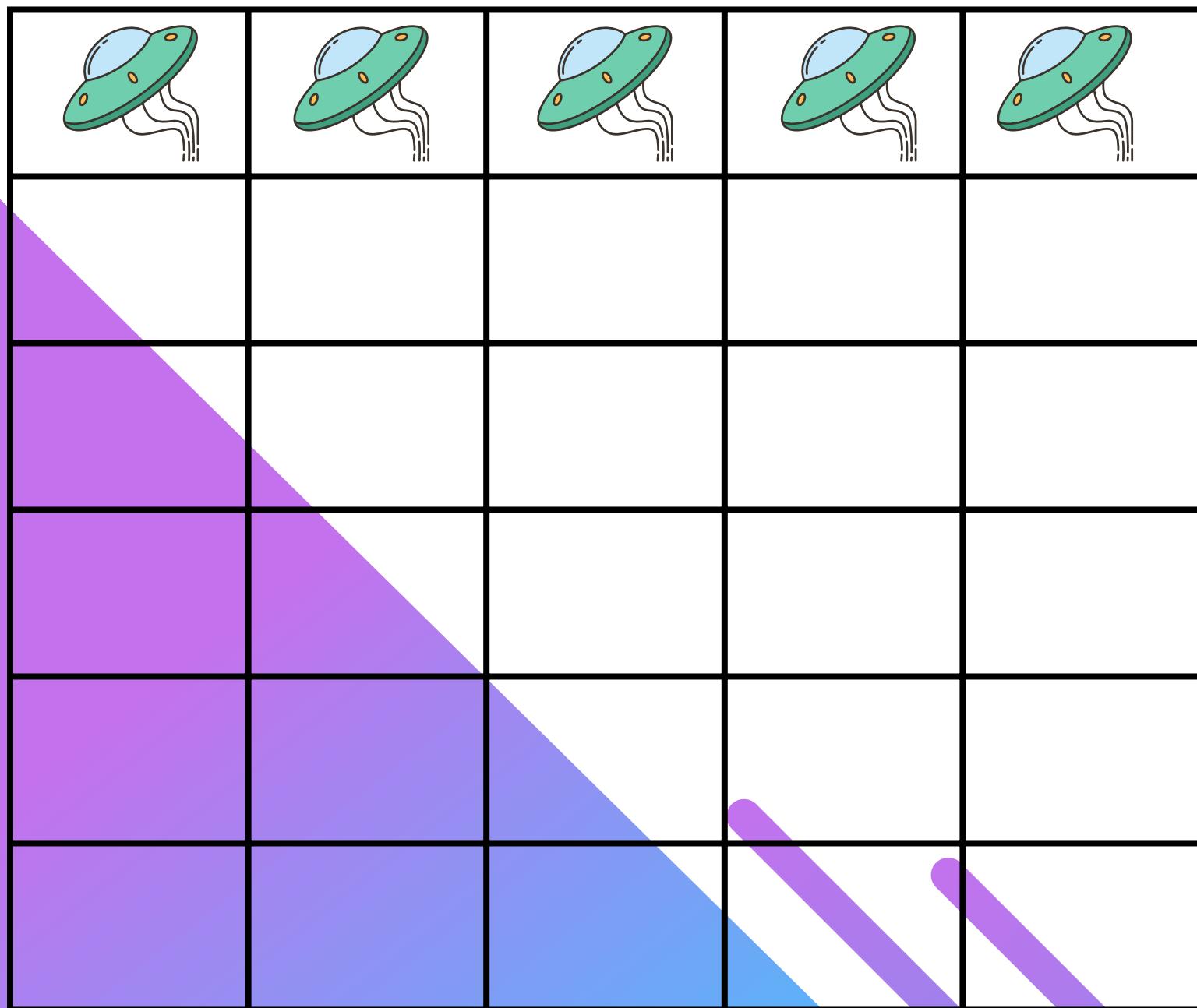
Desarrollo de un sistema multiagente para resolver una tarea cooperativa en un entorno 20x20 dinámicamente cambiante. Donde los agentes no pueden moverse hacia la misma posición.





Problema

20 x 20



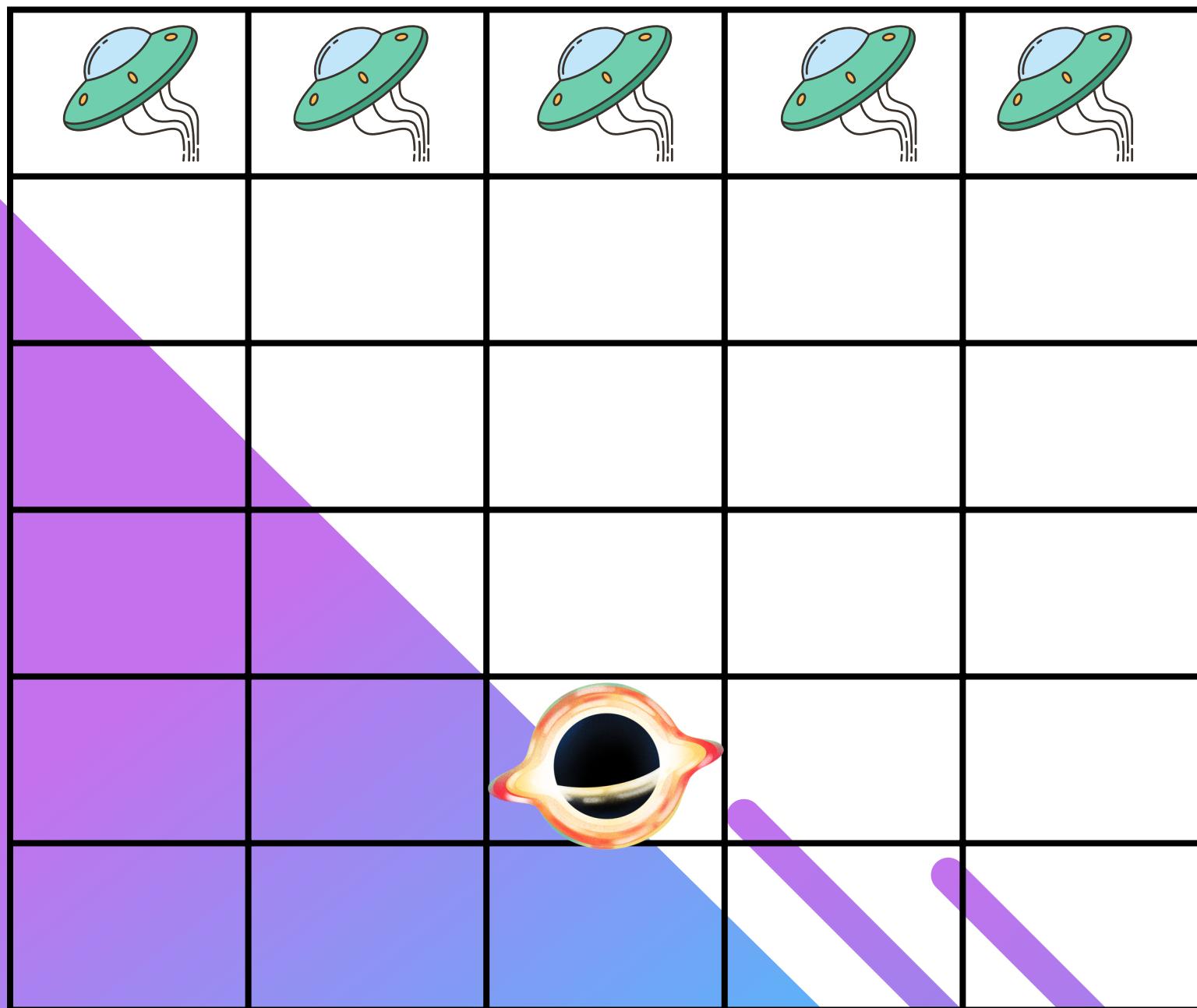
Desarrollo de un sistema multiagente para resolver una tarea cooperativa en un entorno 20x20 dinámicamente cambiante. Donde los agentes no pueden moverse hacia la misma posición.





Problema

20 x 20



Desarrollo de un sistema multiagente para resolver una tarea cooperativa en un entorno 20x20 dinámicamente cambiante. Donde los agentes no pueden moverse hacia la misma posición.

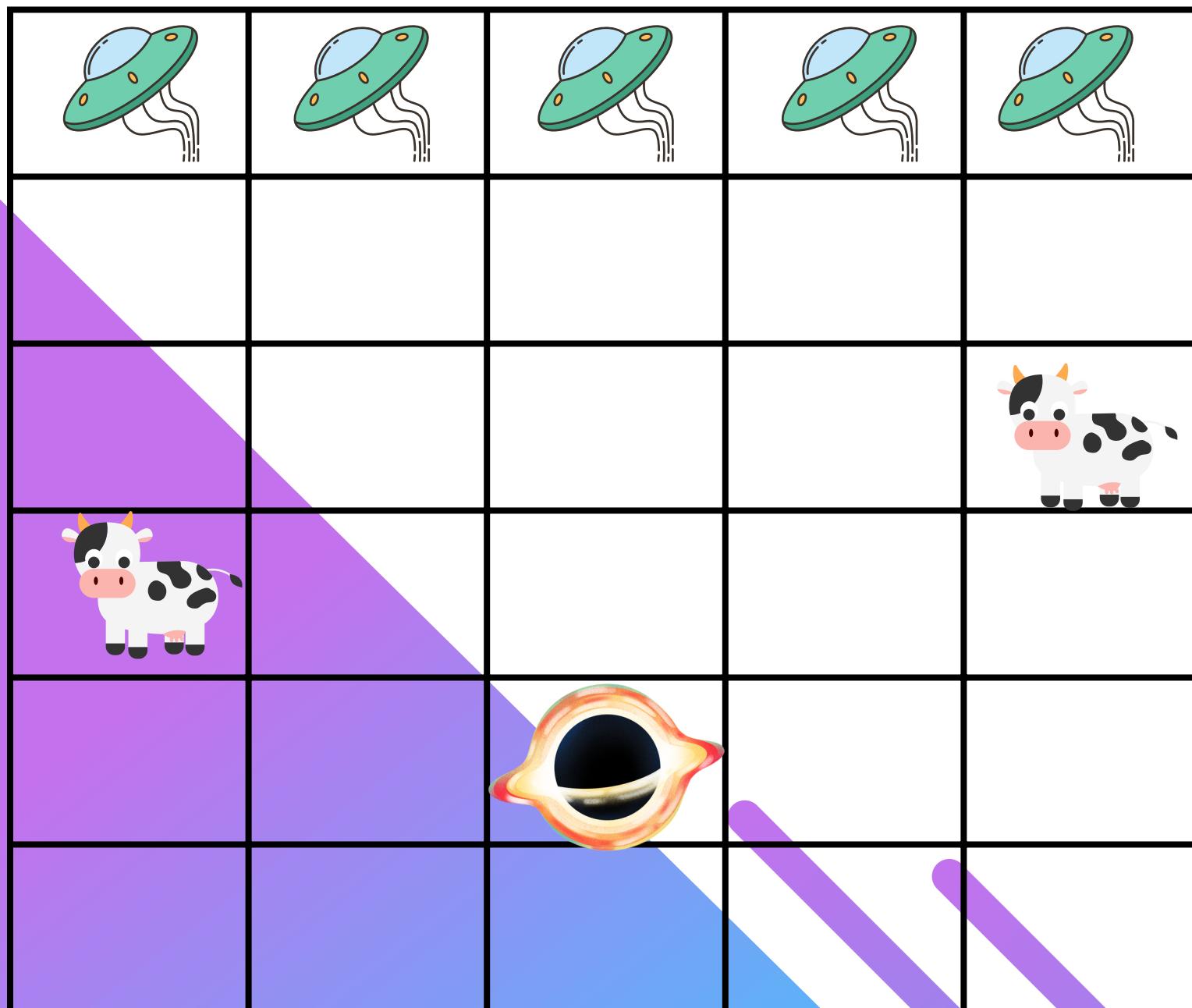




Problema

20 x 20

Max: 47



Desarrollo de un sistema multiagente para resolver una tarea cooperativa en un entorno 20x20 dinámicamente cambiante. Donde los agentes no pueden moverse hacia la misma posición.

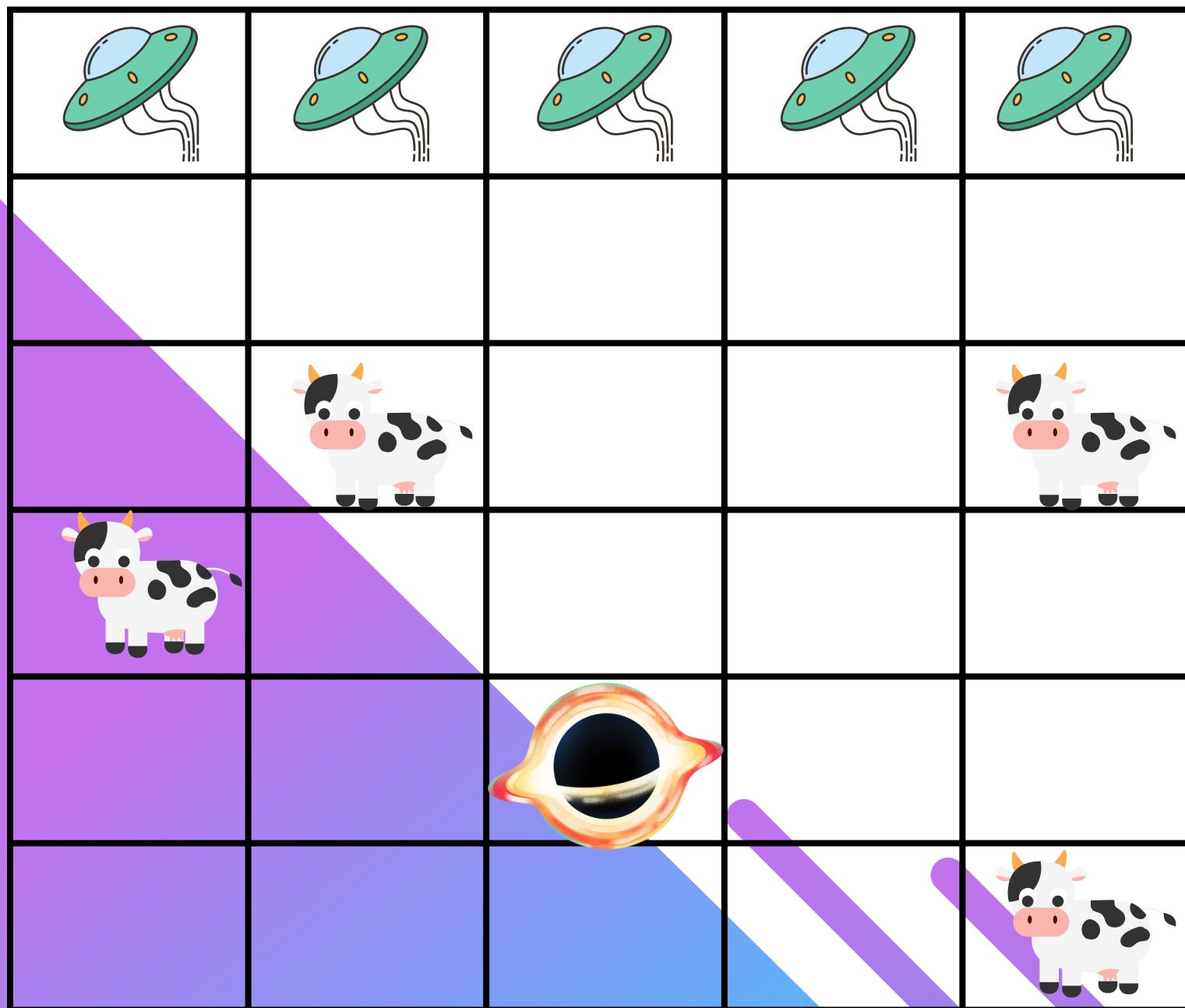




Problema

20 x 20

Max: 47



Desarrollo de un sistema multiagente para resolver una tarea cooperativa en un entorno 20x20 dinámicamente cambiante. Donde los agentes no pueden moverse hacia la misma posición.

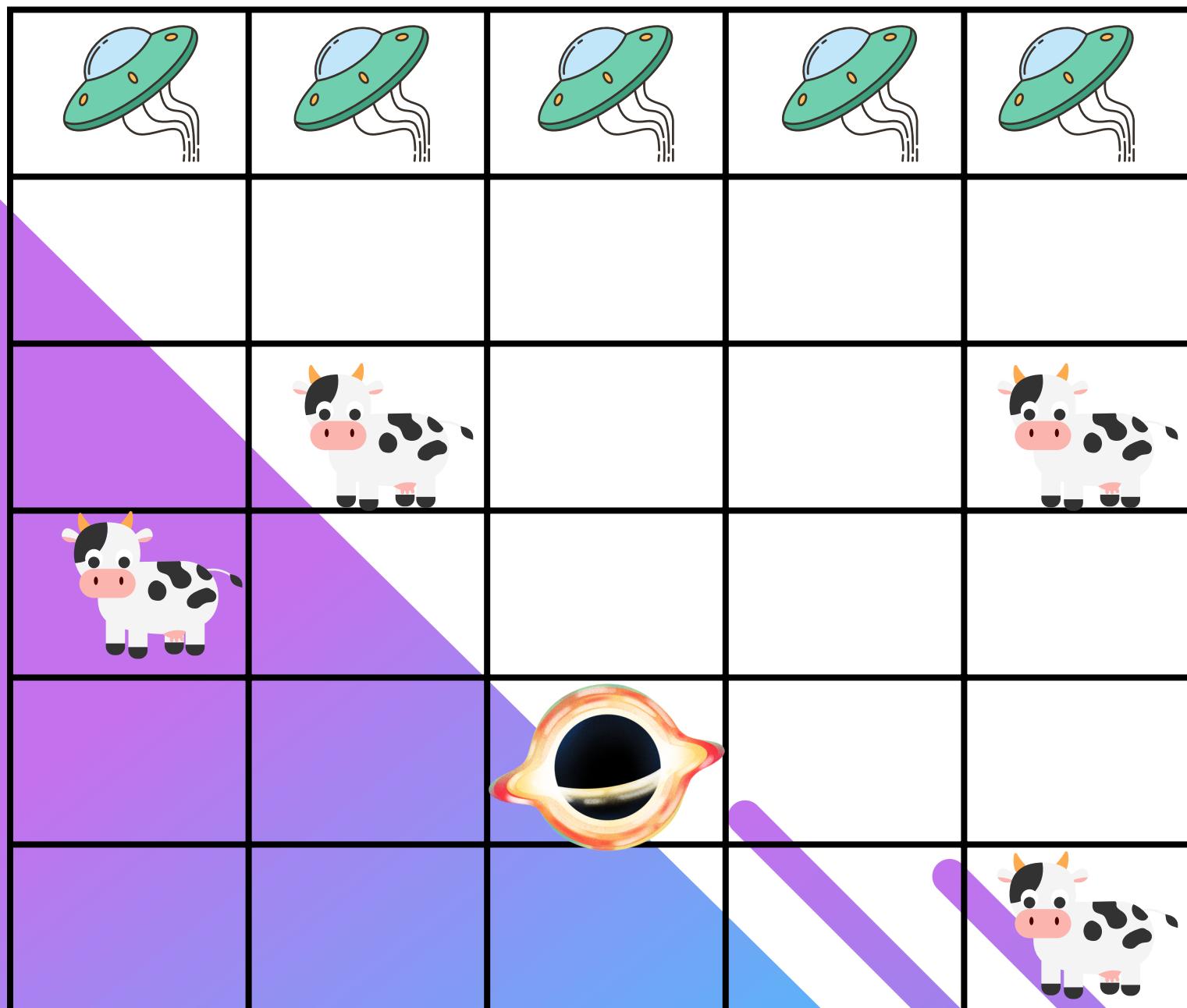




Problema

20 x 20

Max: 47



Desarrollo de un sistema multiagente para resolver una tarea cooperativa en un entorno 20x20 dinámicamente cambiante. Donde los agentes no pueden moverse hacia la misma posición.

Lista de comida

[(5,3)]



Solución

01

Buscar toda la comida

Los agentes se posicionan en zonas estratégicas para recorrer toda la matriz de forma organizada y poder saber donde esta la comida

02

Buscar la comida mas cercana

Los agentes ven cuál es la comida que les queda más cerca

03

Ir por la comida

En este paso el agente va por la comida siguiendo el camino más corto y evitando colisiones con los otros agentes

04

Llevarla al depósito

Ahora se lleva la comida al depósito, siguiendo el camino más corto y evitando colisiones con otros agentes



Modelo Grafico

Mapa:

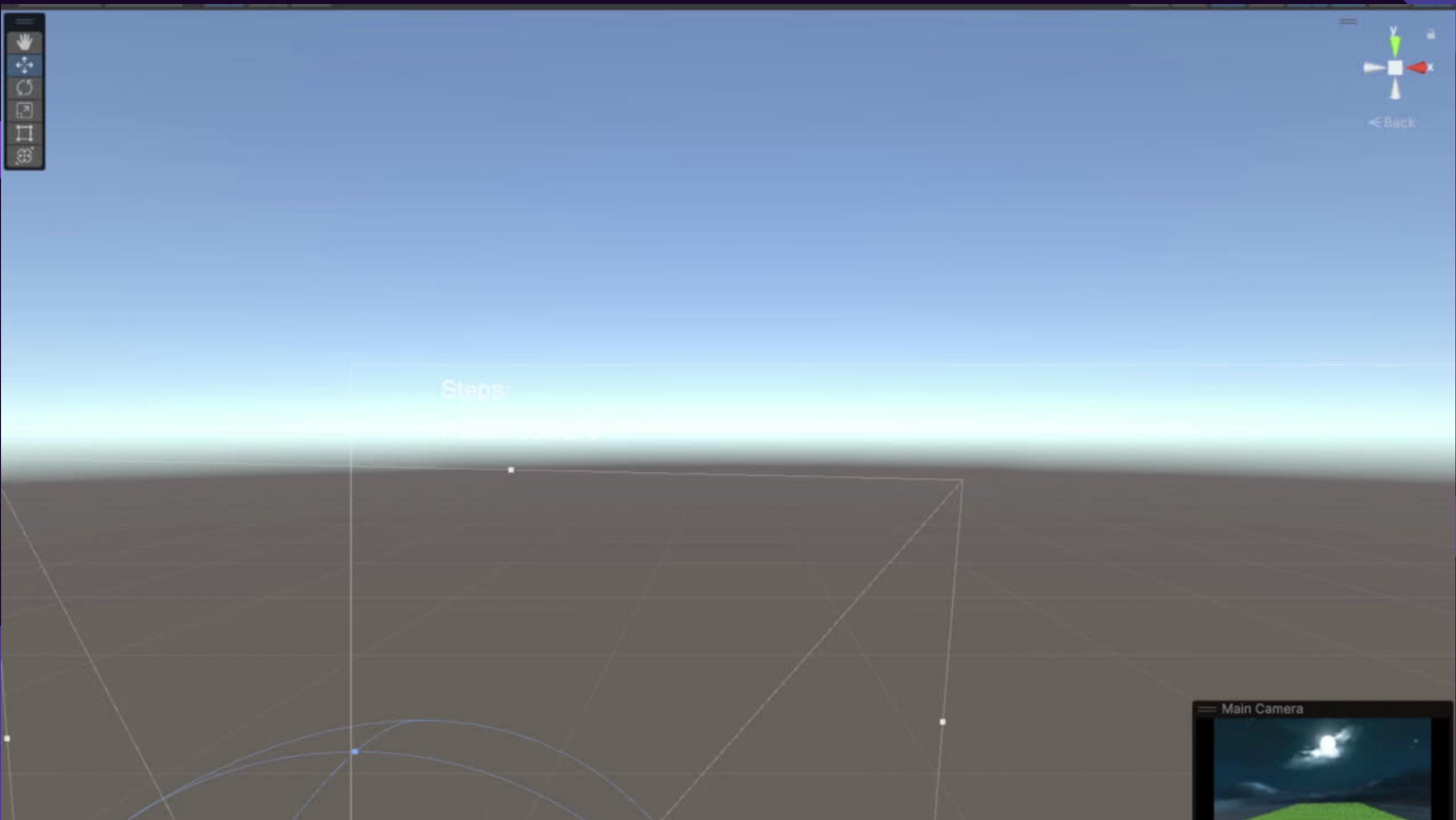
- Terreno boscoso
- Noche
- Movimiento del cielo
- Luz global y direccional en sintonia con la luna

Agentes:

- Naves espaciales
- Separadas por color
- Se activan y desactivan rayos para saber cuando esta cargando comida



Video





Modelo utilizado

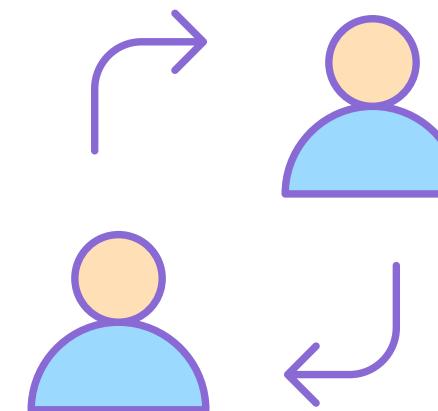
Snake pattern

Usamos un sistema para poder recorrer toda la matriz de una forma ordenada



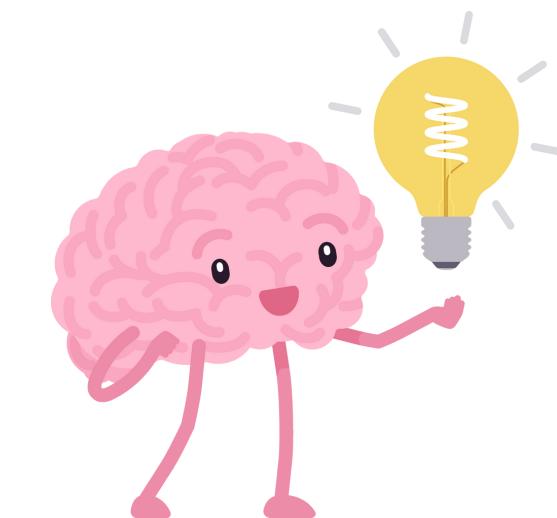
Cambio de rol

Los 5 agentes inician como exploradores, una vez encontrada toda la comida se convierten en recolectores.



Movimientos óptimos

Los agentes siempre van a ir por la comida que les quede más cercada y van a tomar el camino más corto.





Interacción variables





Agente Explorador

- **Todos inician como explorador:**
 - Se dividen el piso entre la cantidad de agentes
 - Se asignan zonas a buscar
 - Buscan en sus zonas asignadas, se verifica si hay comida/depósito donde se encuentra y se comunica a los demás.
 - Al encontrar toda la comida pasan a recolectores

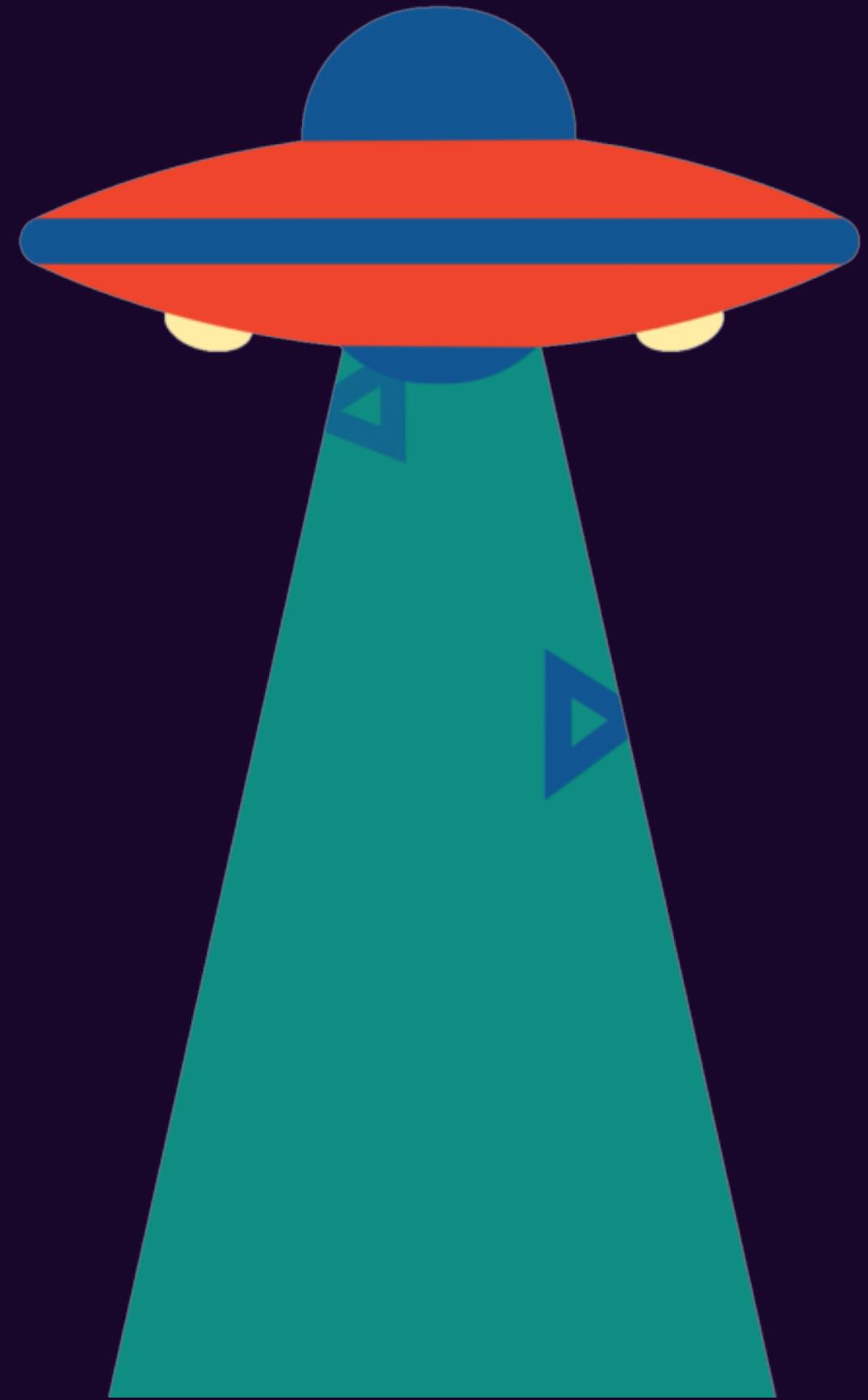




>>>

Agente Recolector

- **Se convierten en recolector**
 - Se revisan las posiciones de la comida y encuentran la más cercana
 - Se revisan los posibles pasos (celdas vacías adyacentes) y se mueve a la más cercana a la comida.
 - Al tener comida, se busca cual de sus posibles pasos lo acerca más al depósito y se mueve .





>>>

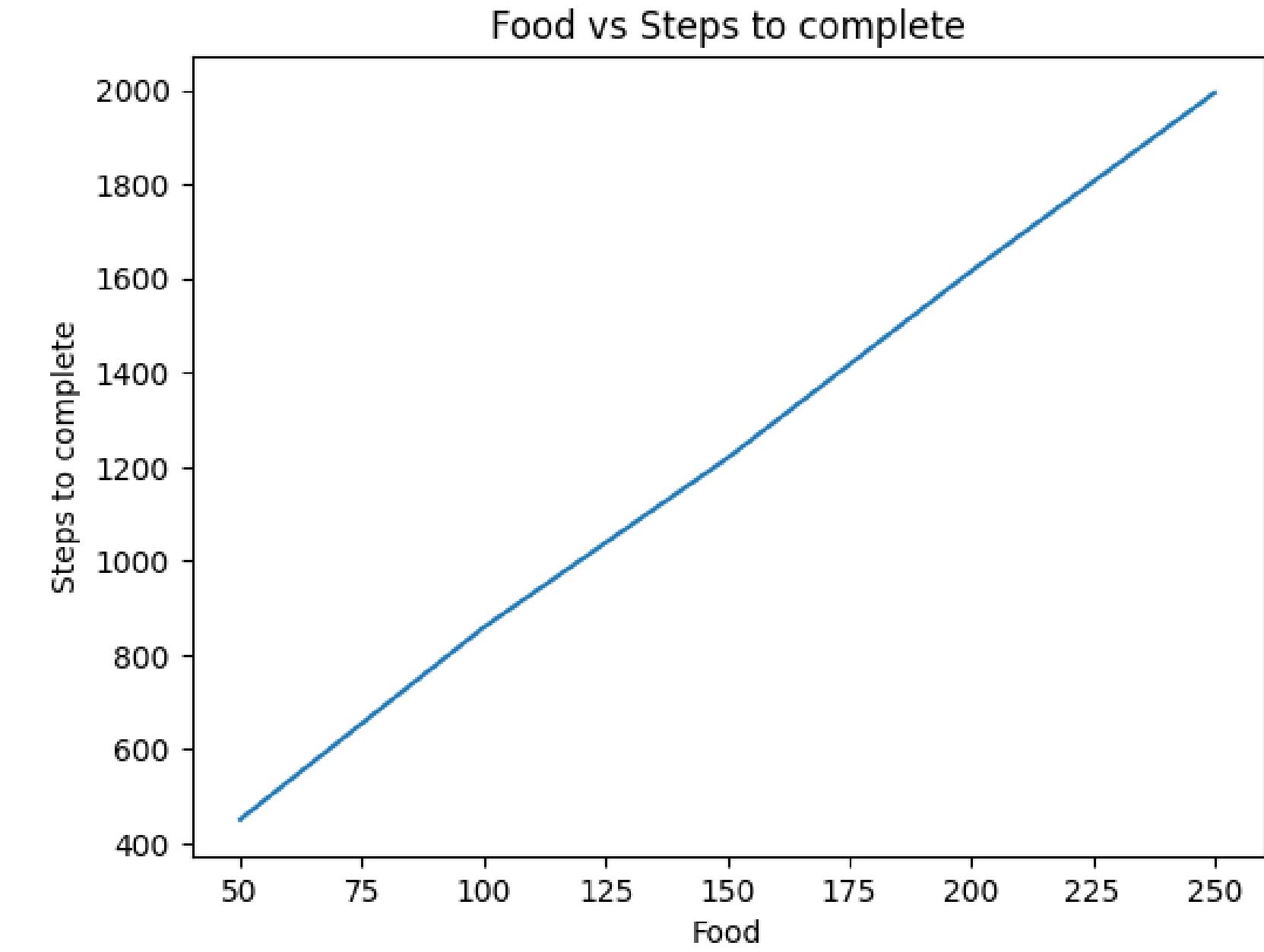
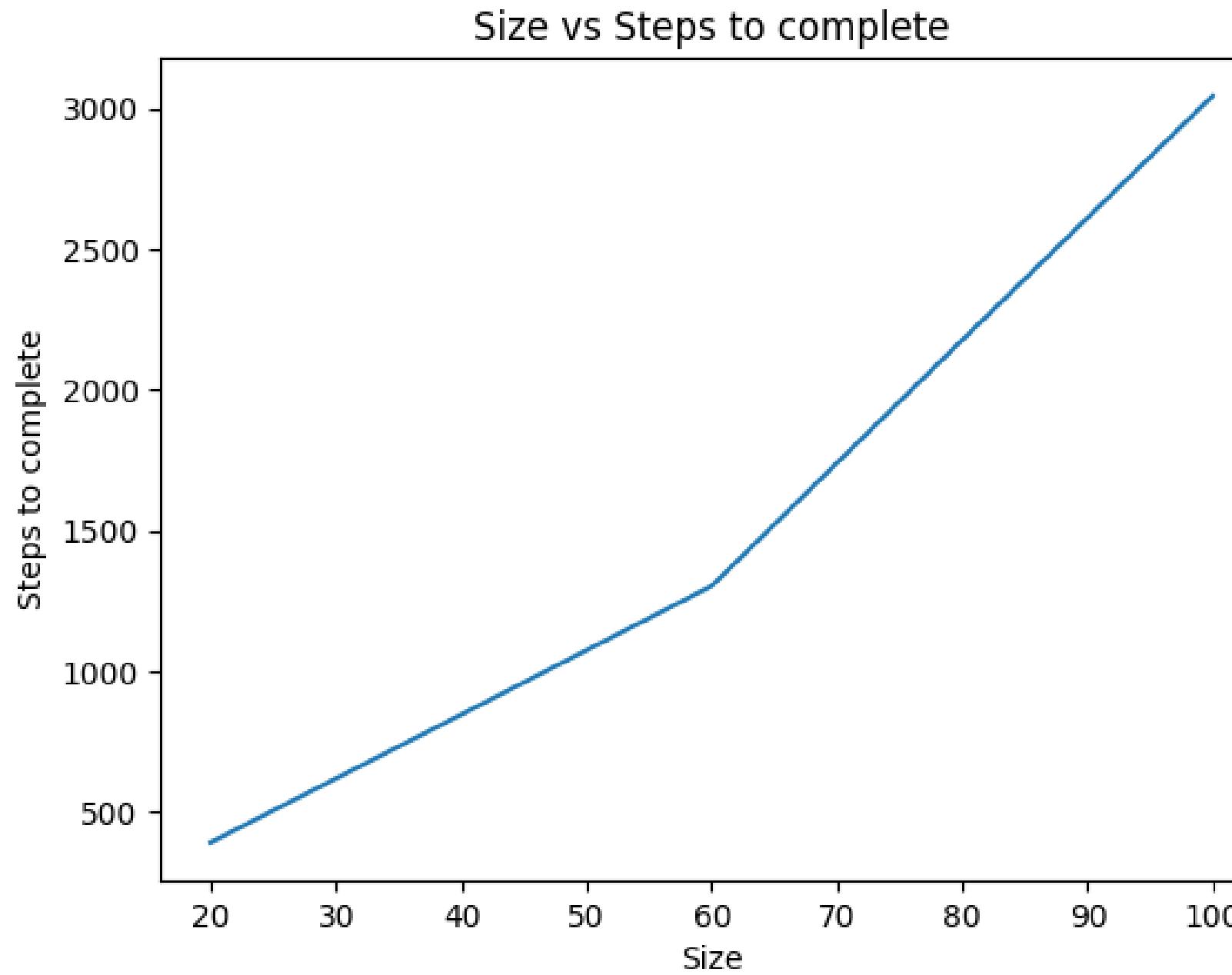
Comunicación de agentes

- Existen 2 propiedades a las cuales los agentes tienen acceso:
 - Coordenadas del deposito
 - Lista con posiciones de comida
- Al encontrar el depósito o comida se actualizan estas.





Cambios de variables

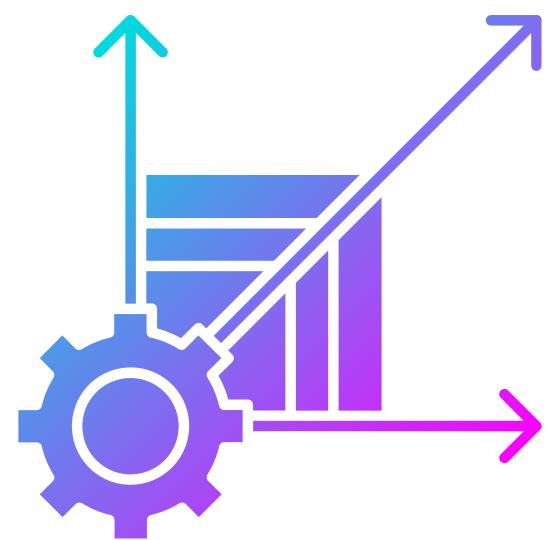




Ventajas de nuestra solución

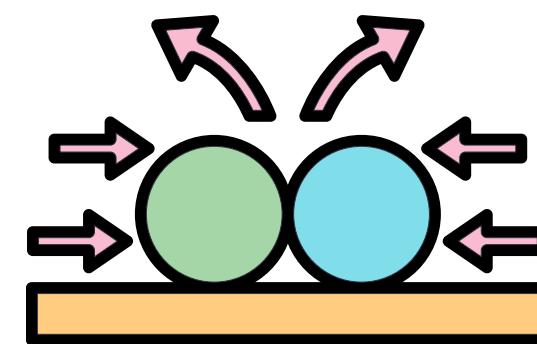
Escalable

Si se aumenta el número de comida y del grid, funciona bastante bien el algoritmo.



Evitas colisiones

Se evitan colisiones, al seguir una ruta predeterminada.



Comunicación asíncrona

Los agentes tienen una comunicación asíncrona, pues una vez que los exploradores detectan toda la comida se cambian los roles.



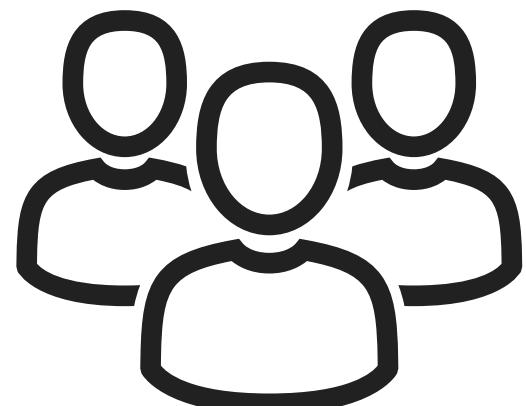


>>>

Desventajas de nuestra solución

Aumento en número de agentes

Entorpecimiento del movimiento de agentes si no hay un buen ratio: agentes-tamaño de grid



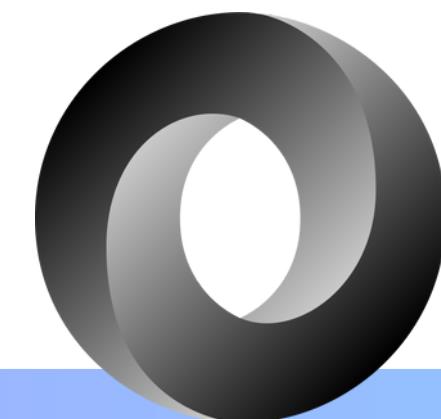
Dependencia de la suerte

Obligación de primero encontrar toda la comida y luego cambiar de roles.



Problemas JSON y Unity

Problemas en Unity por no mandar toda la información requerido con llave de objeto con información requerida.



¿COMO DIMINUIR/ELIMINAR LAS DESVENTAJAS?



Si la diferencia del número de agentes con el tamaño de grid es muy poca, conviene tener cierto número de "Collectors" y número de "Explorers" en un inicio.

En lugar de solo calcular una comida más cercana a la posición si eres "Collector", calcular varias para tener varias ubicaciones para ir.



Conclusiones

Nos dimos cuenta como es que los sistemas multi-agentes son utilizados para resolver un problema general en un entorno cambiante dinámicamente con ayuda de la cooperación y comunicación por parte de cada agente así como su representación mediante gráficas computacionales para plasmar la modelación, obteniendo así una mejor visualización de la problemática resuelta

GRACIAS

