

Práctica 2: Equipo de Defensa

Agentes Inteligentes (3C021)

David Oltra Sanz

Diana Bachynska

1 Introducción

La estrategia que hemos desarrollado se corresponde con la del equipo de defensa: el team Axis. El equipo está formado por 6 soldados, 2 médicos y 2 operativos. De los 6 soldados, 1 de ellos será elegido como general, siguiendo una estrategia que explicaremos a continuación. El general se encargará principalmente de coordinar las comunicaciones y acciones entre los componentes del equipo, así como de establecer una formación al inicio de la partida. Esta formación consistirá en mandar lo más cerca posible de la bandera a los operativos y seguidamente a los soldados para que protejan a los médicos y fieldops, y principalmente a la bandera. Los médicos y el general se quedarán en la base.

Para cumplir con los requisitos del proyecto, hemos implementado varias mejoras en el comportamiento de los agentes. Estas mejoras incluyen la gestión del fuego amigo, la creación de un nuevo servicio denominado "general" y la implementación de nuevas acciones internas dentro del archivo Python. Estas acciones permiten seleccionar al general al inicio de la partida y en caso de baja, designar al médico o al operador de campo más cercano para asistir al soldado en peligro y más acciones comentadas que podremos encontrar en los ficheros. Por tanto, los ficheros implementados son:

- medic.asl
- fieldop.asl
- soldado1.asl

2 Fuego Amigo

Para abordar el problema del fuego amigo, hemos desarrollado una función en el archivo `troopconfig1.py` que calcula la posición relativa de un agente, su amigo y un enemigo. Si la distancia entre el agente y su amigo es inferior a 2.5 unidades, la función retorna `True`, señalando que el amigo está demasiado cerca y sería peligroso disparar. Por otra parte, la función evalúa el ángulo formado entre el agente y su amigo respecto al enemigo, si este ángulo es menor de 15 grados, indica que el amigo está directamente en la línea de tiro, por lo que también se recomienda no disparar. En cualquier otra situación, la función devuelve `False`, indicando que no existe riesgo de dañar al amigo.

3 Elección del General

El criterio de elección de general que decidimos es que el general sea el soldado más alejado de la bandera.

Al principio de la partida cada agente tiene el bit de general a 1, es decir que puede ser general. Después, determina su posición a la bandera usando el método `distance_to` implementado en python, y envía esta posición a los demás agentes. Comparamos las posiciones en el método `compararAgentes` para determinar si al agente le llega una distancia a la bandera mayor que la suya. En ese caso el bit de general se pone a 0. Si a un agente no le llega ninguna distancia a la bandera mayor que la suya es porque es el agente más lejos de la bandera, por lo tanto se elige como general.

Si el general no está disponible cuando se le llama, se vuelve a llamar al método de elección de agentes y se repite el proceso.

4 Comunicación con médicos y fieldops

Cuando la vida de un agente se reduce por debajo del 60%, éste solicita al General que le envíe un médico para ayudarlo. Si el general no está asignando un recurso, empieza el proceso de asignar un médico creando la creencia generalDecidiendo y no puede empezar otro proceso de ayuda hasta que termine el actual. El general obtiene la lista de médicos disponibles y les envía una petición de ayuda. Los médicos responden enviando su posición y el general calcula cual es el médico que está más cerca del soldado que ha pedido ayuda. Entonces acepta la petición de ayuda de éste médico y rechaza la de los demás. El médico que ha sido contratado va a la posición del soldado a curarlo.

La comunicación con los fieldops es equivalente.

5 Posición válida mapa

Dado que al comienzo de la partida, el general asigna a los agentes una posición específica en el mapa, es posible, que en una mapa esa posición si sea transitable, pero si cambiamos de mapa, esa posición no sea válida debido a que hay un obstáculo. Por esta razón, hemos implementado una función interna que verifica si la posición es accesible utilizando el método 'self.map.can_walk(X,Z)'. Si la posición resulta inaccesible, la función redirige automáticamente al agente hacia la bandera.

6 Conclusiones

Decidimos mantener al general y a los médicos en la base, ya que, siendo los miembros más importantes del equipo, es fundamental asegurar su supervivencia. Sin embargo, enfrentamos un inconveniente: la respuesta del médico era demasiado lenta cuando se le solicitaba su ayuda en otro lugar.