

Práctica 2: Agentes deliberativos

En esta práctica el comportamiento reactivo que he utilizado es similar al de la práctica anterior, ya que obtenía un porcentaje mayor al 85% en todos los mapas y me parecía correcto. Un cambio que he hecho es que ahora voy a ignorar los huesos. La razón es porque tenemos sólo 4 huecos en la mochila y siempre dejo un hueco libre para que cuando tenga que coger objetos de la mochila pueda meter el que tengo en la mano y sacar otro. El hueso es el objeto que menos importancia tiene así que he decidido no utilizarlo. Este cambio se aprecia en las líneas 435-515. Cuando tengo un hueso en la baldosa de en frente, hago un giro para esquivarlo. Si es otro objeto, lo cojo.

La otra modificación que he hecho está en las líneas 393-396 y 411-425. Es muy similar a la forma que tengo de ir a por los puntos PK. Tengo unas funciones que me devuelven si puedo detectar un rey cerca. Si tengo un rey cerca y tengo un regalo, voy a ir hacia él hasta situarme delante para entregarle el regalo. Esto podría haberlo hecho de forma deliberativa, pero sería gastar segundos en algo que es muy simple.

Ahora voy a explicar el comportamiento deliberativo. Mi idea es contar el número de turnos que pasan para cambiar de reactivo a deliberativo. Si han pasado más de 200 turnos, intento planificar un camino al regalo si aún no tengo uno y si tengo uno intento planificar un camino a algún rey que tenga guardado en una lista de reyes. En ambos casos, voy a intentar coger el camino más corto haciendo la distancia entre dos puntos.

Voy a intentar seguir el orden del código.

- Líneas 121-124: si tengo un error en el plan que tengo planificado, no ejecuto más el plan.
- Líneas 127-242: si estoy ejecutando el plan y la siguiente acción es avanzar, voy a modificar el plan en caso de que sea necesario. Esto es:
 - Líneas 158-161: si tengo delante un aldeano o un lobo, pongo la acción actIDLE al principio de mi plan para estar parado hasta que pueda avanzar.
 - Líneas 163-236: si tengo un hueso, voy a cogerlo y voy a girarme para lanzarlo y volver a girar para volver a mi posición inicial y así quitarlo de en medio. Si tengo cualquiera de los otros objetos, en caso de que tenga ya ese objeto voy a actuar de la misma forma que con el hueso. Si no tengo el objeto, voy a guardar el objeto que tenga en la mano y voy a cogerlo.
 - Líneas 238-244: si voy al agua y tengo el bikini y no estoy sobre bosque, si no tengo el bikini activo voy a ponérmelo.
 - Líneas 246- 252: lo mismo que antes, pero esta vez con el bosque y las botas.
 - Líneas 254-256: si voy a ir hacia el agua y no tengo el bikini, voy a abortar mi plan.
 - Líneas 258-260: si voy a ir hacia el bosque y no tengo las botas, voy a abortar mi plan.
 - Líneas 262-264: si tengo delante un precipicio o un muro, voy a abortar mi plan.
 - Líneas 266-272: si no pasa nada de lo anterior, voy a suponer que no hay error en el plan y puedo seguir con él.
- Líneas 244-272: si estoy bien situado y soy capaz de ver un rey cerca, voy a guardar la posición en la que se encuentra el rey. Mi idea es tener una lista de todos los reyes con los

que me vaya encontrando, y voy a guardar el punto que corresponda a la baldosa situada al norte, sur, este u oeste del rey. El punto no va a ser guardado en caso de que ya esté en la lista o en caso de que la baldosa tenga agua o bosque, pues si me sitúo en alguno de estos dos y cambio el objeto activo para dar el regalo, voy a morir. Además, tengo una variable *replanificar*. Esta variable me sirve para cuando por ejemplo estoy ejecutando un plan y me encuentro un rey. Este rey seguramente va a estar más cerca que el rey al que estoy intentando ir, entonces voy a indicar que tengo que replanificar para ir hacia este rey que está más cerca.

- Líneas 275-304: aquí va la idea que he comentado al principio. ¿Cuándo voy a hacer un plan para llegar a un regalo? Cuando esté bien situado, no tenga regalo, no esté ejecutando otro plan, haya regalos en el mapa, mi lista de reyes no esté vacía (no le veo sentido a ir a por un regalo si aún no tengo algún rey localizado) y el número de turnos sea mayor a 200. Mi plan va a llegar hasta el regalo que se encuentre más cerca. Para ello, voy a calcular la distancia entre dos puntos: desde mi posición original hasta las posiciones de todos los regalos. Voy a calcular el pathFinding para el regalo que tenga una distancia menor.
- Líneas 306-334: aquí hago el plan para llegar a algún rey.y Voy a calcular el camino hasta a un rey en alguna de estas dos ocasiones:
 - Si estoy bien situado, tengo regalo, no estoy ejecutando otro plan, tengo algún rey localizado y el número de turnos es mayor.
 - Si estoy bien situado, tengo regalo, no estoy ejecutando otro plan, tengo algún rey localizado y he indicado que voy a replanificar.En ambos casos lo que voy a hacer va a ser algo parecido a la planificación del regalo. Voy a recorrer la lista de puntos en los que están localizados los reyes y voy a calcular la distancia entre la posición en la que estoy y la posición de todos los reyes para encontrar el rey que más cerca tengo. La idea del segundo punto (la que tiene variable *replanificar*) es la que he explicado en el primer punto: si estoy ejecutando un plan para ir hacia un rey y me he encontrado por el camino otro rey, este rey estará más cerca y me interesará entregar el regalo a este.

Algoritmo pathFinding

He utilizado la búsqueda en anchura. Para este algoritmo tengo dos struct:

- Struct estado: va a representar un estado con una fila, una columna y una orientación.
- Struct nodo: va a representar los nodos del “árbol” que voy a recorrer para encontrar una solución. Va a conter un estado y una lista de acciones.

Voy a tener dos listas de nodos: una lista de abiertos, que son los nodos que me quedan por explorar, y una lista de cerrados con los nodos que ya he explorado.

A partir del nodo inicial (la posición origen en la que estoy situado) voy a crear 3 hijos en cada nodo:

- Un hijo va a contener la acción de avanzar.
- Otro hijo va a contener la acción de girar hacia la izquierda.
- El otro hijo va a contener la acción de girar a la derecha.

Cada hijo lo voy a meter en la lista de abiertos siempre y cuando:

- No esté el nodo en la lista de abiertos o cerrados y además la columna y fila sea un número comprendido entre 3 y 96, pues no voy a explorar la posición [-1][5], por ejemplo.
- Además de lo anterior, voy a meter el hijo en caso de que su casilla sea suelo arenoso o pedregoso, una casilla desconocida aún, bosque o sea agua. Para esto, utilizo el mapa que tengo guardado para utilizar la técnica de pulgarcito.
- Por último, para no crear muchos nodos debido a que la eficiencia es el punto débil de este algoritmo, calculo la fila y columna del hijo creado y la resto a la fila y columna de mi destino. Si la distancia está a más de 20 casillas de la distancia que hay entre la fila y columna de origen (el punto en el que empezó el algoritmo), tampoco voy a meter ese nodo, pues me estoy alejando demasiado del punto y puedo suponer que ese camino no va a ser viable.

Esto lo voy a repetir mientras mi lista de nodos abiertos no esté vacía y no haya encontrado la solución, es decir, que el nodo que he cogido de abiertos no tenga el mismo estado que el estado objetivo.

Conclusión

Mi intención es ir a lo seguro. Intento explorar el mapa lo suficiente como para poder hacer un buen plan, y a la mínima en la que mi plan puede fracasar, intento seguir explorando el mapa de forma reactiva para aumentar el conocimiento. Además, intento crear nodos que sean explorables para garantizar que el plan no va a ser un disparate. La idea final es coger un regalo, explorar el mapa de forma reactiva y luego buscar a un rey para dar el regalo.