

Agentes Reactivos/Deliberativos

David Cabezas Berrido

29 de abril de 2019

1. Búsqueda en anchura

Ya implementada, usa una pila (LIFO) para expandir primero los últimos nodos que se almacenaron.

2. Búsqueda en anchura

He utilizado el código de la búsqueda en profundidad, modificando la estructura de datos. He cambiado la pila (LIFO) por una cola (FIFO), de forma que ahora el árbol de búsqueda se recorre en anchura.

3. Búsqueda en costo uniforme

He vuelto a reutilizar el código, pero cambiando el criterio para elegir los nodos a expandir. En lugar de LIFO o FIFO, selecciono los nodos por una heurística, si n es un nodo:

$$H(n) = n.coste \cdot d_1(n.estado, destino)$$

donde el coste de un nodo es la penalización por terreno del camino recorrido hasta el momento (desde la posición actual del jugador), y el otro término del producto es la distancia de Manhattan desde la posición del nodo hasta el destino.

Selecciono para expandir el nodo n que minimice el valor de $H(n)$, así doy importancia la penalización del suelo pero evito que en el árbol de búsqueda se exploren caminos que alejan al jugador del objetivo por el hecho de tener una baja penalización por suelo.

Es posible que el camino encontrado no sea el óptimo en lo que a penalización del terreno se refiere, pero el tiempo de cómputo empleado en la búsqueda es entre 10 y 100 veces menor. Luego sacrificamos calidad del camino encontrado para mejorar rendimiento.

4. Reto

Para el comportamiento reactivo/deliberativo, he diseñado la siguiente solución:

Mapa auxiliar:

He creado un mapa auxiliar del doble de lado, Belkan siempre empieza en el centro y utiliza este mapa hasta encontrar una casilla de referencia, en ese momento copia en el mapa original lo que lleve explorado del mapa auxiliar y pasa a utilizar el mapa original.

Con utilizar el mapa me refiero a memorizar el terreno, comprobar si hay obstáculo o consultar el coste de una casilla. Los planes los hace con las coordenadas relativas al mapa que esté usando.

Exploración:

A medida que Belkan se mueve, va memorizando en el mapa (tanto el auxiliar como el original) lo que ve en los sensores de terreno.

Búsqueda de casilla de referencia:

Tras cada movimiento, Belkan comprueba si hay alguna casilla de referencia en sus sensores de terreno y, en caso afirmativo, traza un plan hacia ella y lo sigue hasta alcanzarla. Una vez ubicado, Belkan no vuelve a realizar esta comprobación.

Comportamiento sin referencia:

Cuando Belkan no ha encontrado aún una casilla de referencia, se mueve aleatoriamente por el mapa. He dado mayor probabilidad a la acción de avanzar hacia delante para evitar que Belkan permanezca siempre en la misma zona.

Comportamiento con referencia:

Cuando Belkan conoce su posición, traza un plan hacia el destino, considerando que las casillas inexploradas no son obstáculos. Hay que darle una penalización por terreno a las casillas inexploradas: una alta penalización hará que Belkan explore menos el mapa y priorice los caminos ya conocidos, una baja penalización provoca que Belkan explore mucho el mapa (lo que nos beneficiará durante el resto de la partida) pero puede dar lugar a caminos peores o sin salida.

Abortar el plan:

Cuando Belkan está siguiendo un plan que incluye casillas desconocidas, es posible que esas casillas acaben siendo muros o precipicios y no sea posible seguir. En este caso Belkan aborta el plan y calcula uno nuevo, ahora con más información sobre el mapa.

Aldeanos:

Cuando la siguiente acción de Belkan es avanzar pero un aldeano no se lo permite, Belkan deja de consumir acciones del plan y mantiene una agradable conversación con el aldeano (espera) hasta que éste decide seguir su camino.