Memoria de la 2ª Práctica de Inteligencia Artificial

Los mundos de Belkan

-

Nivel 1

Se realiza una *busqueda en anchura* desde la casilla inicial(donde aparece el personaje) hasta que encuentra el destino pasado como argumento.

Se visitan la casilla actual y las cuatro adyacentes (N, S, E, O). Estas casillas se introducen en una cola de manera que se van atendiendo en el mismo orden en el que son introducidas. Una vez sacadas de la cola se realiza el mismo procedimiento en ellas: se registran como visitadas las casillas adyacentes y se añaden a la cola para ser tratadas.

A la vez que se visita una casilla adyacente se registra tambien a traves de que casilla se ha accedido a ella, es decir, su anterior. De esta manera es como logramos encontrer el camino desde nuestra posición hasta el objetivo o, realmente, desde el objetivo a nuestra posición: una vez alcanzado el objetivo recorremos las casillas previas que hemos ido guardando hasta coincidir con la de origen. Este recorrido de casillas es el camino viable entre la posición origen y destino.

La orientación del personaje la escogemos según la orientación guardada en los estados de las casillas. Esto lo hacemos teniendo en cuenta la posición de su casilla previa. Por ejemplo: si la casilla adyacente es (3, 4) y desde la que hemos accedido es la (3, 3) hemos aumentado una columna por lo que la orientación, según la disposición de la matriz, es ESTE (avanzamos hacia la derecha). Esta orientación será la que tome también el personaje.

-

Nivel 2

El algoritmo de búsqueda es el mismo que el del ***nivel 1*** pero en este caso, al haber aldeanos que pueden resultar un obstáculo, debemos decidir cómo reaccionar ante ellos. En mi caso he optado simplemente por *esperar a que el aldeano se aparte* del camino trazado.

-

Nivel 3

En este último nivel en el que no conocemos nuestra posición en el mapa he aprovechado el algoritmo de búsqueda en anchura para ir en busca del punto ***PK***:

Primero traslado el vector de visión que tiene nuestro personaje a una matriz 7x7 y busco un camino en anchura hasta la esquina izquierda y derecha de su visión. Voy altenando cada dos iteraciones a que esquina debe ir. Esto crea un movimiento de serpenteo que permite ocupar mayor terreno. Además cada 20 de estas iteraciones se añade una más a una de las esquinas provocando un cambio de dirección y evitando que pueda quedar en un bucle. El movernos hacia un ***PK*** aprovechando el algoritmo de ***búsqueda en anchura*** nos permite salir de recovecos.

Una vez hallado el punto ***PK*** volvemos a usar la ***búsqueda en anchura*** en la matriz del mapa original y en cada movimiento vamos pintando lo que ve nuestro personaje mediante la función `pintarMapa()` que nos permitirá usar la búsqueda en anchura más eficientemente.

El algoritmo crea un recorrido ajustándose al mapa ya explorado pero en las zonas todavía no vistas traza el camino más corto (en linea recta). En caso de que no sea viable esta trayectoria se volverá a llamar al algoritmo de búsqueda en el instante justo donde no pueda continuar avanzando para que actualice el camino con la porción nueva de mapa explorado.