

Ingeniería del conocimiento: IA

Memoria IA: Pathfinding

PRÁCTICA 1

Jorge Bárcena Lumbreras
Miguel Ángel Gil Martín

ÍNDICE

ÍNDICE	1
INTRODUCCIÓN	2
Análisis de resultados	2
PROBLEMAS ENCONTRADOS	4

INTRODUCCIÓN

Esta introducción esta rehecha tras la presentación de la practica al profesor. Podemos dividir la practica en 2 partes.

En primer lugar, vamos a explicar la primera parte.

ALGORITMOS OFFLINE

Hemos realizado el algoritmo de A* y de A Ampliada, para ello tras la reunión con Luis, nos comentó algunos errores que había en el algoritmo. En el de A*, el problema surgía de que al final, cuando ya se había encontrado la meta, no cogía esos nodos y creaba una ruta, sino que simplemente los cogía de la lista cerrada, por eso fallaba el algoritmo. En cuanto cogimos la ruta y se la mandamos al jugador, el algoritmo funciona correctamente.

ALGORITMOS ONLINE

Para realizar este apartado hemos empleado un algoritmo de búsqueda ampliada por horizonte. La idea del algoritmo es que busque una serie de objetivos, en este caso, los enemigos, y que vaya a por cada uno de ellos. Para hacer esto, hará una búsqueda por amplitud cada x segundos, establecidos por el jugador, tras llegar a esa meta, establecida, volverá a hacer un repath hasta el objetivo, así hasta que consiga llegar a él. Cuando lo consiga se establecerá un nuevo objetivo y se irá hacia él.

Análisis de resultados

ALGORITMOS OFFLINE

Para realizar una comparación entre los dos algoritmos que hemos realizado, a continuación, vamos a poner una tabla comparativa entre los dos algoritmos y los nodos que se han expandido en cada uno, en función de una semilla.

SEMILLA	A*	A AMPLITUD
199	1107	317
245	EXCESIVOS NODOS	401
898	NO EXISTE SOLUCION	NO EXISTE SOLUCION
-555	140	109
458	13863	361
986	7252	514
1235	6992	319
2001	8	12
14	26183	583
951	28	180

999	32	25
1235	6992	319
456	36	29
789	8	12
321	308	283
6544	19137	322
987	NO EXISTE SOLUCION	NO EXISTE SOLUCION
444	278	210

Tras estos datos llegamos a la conclusión de que nuestro algoritmo de A* necesita muchos más nodos que el de Amplitud para obtener una solución, por lo que creemos que utilizaríamos el algoritmo de búsqueda en Amplitud. La única ventaja que podemos observar con estos resultados es que el algoritmo de A* es mejor en casos en los que la meta esta muy cerca del jugador.

ALGORITMOS ONLINE

Para realizar el análisis de este algoritmo de búsqueda en amplitud online vamos a fijarnos en la semilla, las veces que recalculamos ruta y los enemigos que hay en la escena:

SEMILLA	REPATH	ENEMIGOS
555	596	3
555	16	1
2019	559	3
2019	157	1
736	69	3
736	17	1
11235812	439	3
11235812	158	1

Tras estos datos vemos que los recálculos de ruta son directamente proporcionales al numero de enemigos que hay en la escena ya que recalcula ruta cada cierto tiempo y si la baldosa objetivo no es caminable vuelve a recalcular, por lo que, si en el momento en el que fija a un enemigo, este está dentro de un muro, el algoritmo estará recalculando ruta hasta que el enemigo salga del muro.

PROBLEMAS ENCONTRADOS

ALGORITMOS OFFLINE

Tras la reunión con Luis, conseguimos solucionar muchos de los problemas que nos daban los dos algoritmos. En el de A^* , el problema surgía de que al final, cuando ya se había encontrado la meta, no cogía esos nodos y creaba una ruta, sino que simplemente los cogía de la lista cerrada, por eso fallaba el algoritmo. En el algoritmo de amplitud el problema surgía a la hora de que comprobábamos que el nodo no estuviera ya en la lista cerrada, cuando se podía dar el caso que, si que lo fuera, fuera el mismo nodo, pero habiendo llegado desde distintas direcciones.

ALGORITMOS ONLINE

El principal problema es a la hora de buscar los objetivos, ya que cuando eliminamos el objetivo conseguido, en ocasiones, con cambia de objetivo por lo que se queda en un estado de bucle infinito, nos dimos cuenta que este bucle infinito se debía a que en el momento en el que fijó a un enemigo, este estaba dentro de un muro y como el jugador nunca llega a las coordenadas marcadas, no recalcula la ruta, para solucionarlo, añadimos una condición por la que comprueba si el objetivo es alcanzable y si no lo es, recalcula, esto genera un problema de rendimiento ya que, el algoritmo dependiendo de la situación puede llegar a realizar una cantidad ingente de recálculos de ruta.