

INTELIGENCIA ARTIFICIAL

CURSO 2022-23

PRACTICA 2: Repertorio de preguntas para la autoevaluación de la práctica 2.

APELLIDOS Y NOMBRE	Hoces Castro, José Alberto		
GRUPO TEORÍA	A	GRUPO PRÁCTICAS	A2D

Instrucciones iniciales

En este formulario aparecen preguntas que requieren breves explicaciones relativas a cómo el estudiante ha hecho algunas partes de esa implementación y qué cosas ha tenido en cuenta.

Enumera los niveles presentados en su práctica (Nivel 0, Nivel 1, Nivel 2, Nivel 3, Nivel 4):

0,1,2 y 3

Nivel 1-Anchura para el agente sonámbulo

- a ¿En qué se diferencia desde el punto de vista de la implementación el algoritmo que has usado en este nivel en relación al del nivel 0? (enumera los cambios y describe brevemente cada uno de ellos)

- 1) Tuve que crear un nuevo estado stateN1 que es idéntico a stateN0 salvo el operador <, donde también se comparan los valores de la fila, columna y orientación del sonámbulo.
- 2) En cuanto a la anchura de este nivel, al igual que antes, se para cuando entra la solución en abiertos. Como en este caso queremos llevar al sonámbulo al objetivo, solo se comprueba que la solución haya entrado a abiertos cuando se genera un nuevo nodo para la acción actSONFORWARD.
- 3) También tuve que crear una función llamada VeoSonambulo que me indicase a partir de la posición del jugador y del sonámbulo si este último estaba en su vista (los sensores en este caso no se podían usar). Por lo tanto, si se ve al sonámbulo, se generan 6 nodos hijos, y en caso contrario, 3.

Nivel 2-Dijkstra para el agente jugador

- a ¿Qué es propio de este nivel que no tuviste que tener en cuenta en los niveles anteriores? (enumera los cambios y describe brevemente cada uno de ellos y que han implicado en la implementación)

- 1) A diferencia de los niveles 0 y 1, ahora se aplica un algoritmo de coste uniforme, por lo que ya no se puede dar por hecho que si una solución entra en abiertos, esta sea la óptima. Por ello, ahora se termina cuando la solución entra en cerrados (es decir, cuando el current_node que se acaba de sacar de frontier es justo solución).

2) StateN2: Es igual que el stateN0 pero añadiendo dos variables booleanas que indican si el jugador posee el bikini o las zapatillas (no puede tener ambos items a la vez). Entonces, en los operadores == y < que había en el nivel 0 tenemos que añadir las comparaciones de estas dos variables nuevas.

3) A diferencia de los dos niveles anteriores, frontier pasa a ser una cola con prioridad que usa el operador < de nodeN2, que está construido sobre stateN2. De esta forma, cuando se saca un elemento de frontier, sabemos que siempre se saca el nodo que tiene mayor batería restante.

4) He creado una función llamada CosteAccion para calcular la batería que se le va a restar a la batería de current_node para crear cada nodo hijo (el coste se calcula con el tipo de terreno sobre el que ya se estaba).

5) Explored (cerrados) pasa a ser un set de stateN2, ya que para ordenar los nodos una vez entrados en cerrados solo me interesa su estado y no se van a revisar. Fue una forma de que no me chocasen los operadores < de nodeN2 para la cola con prioridad con el operador < de stateN2.

- b ¿Has incluido dentro del algoritmo de búsqueda usado en este nivel que si pasas por una casilla que da las zapatillas o el bikini, considere en todos los estados descendientes de él el sonámbulo o el jugador tiene las zapatillas y/o el bikini? En caso afirmativo, explicar brevemente cómo.

Sí, esto lo controlo cuando al generar los hijos, hago que su estado tome el valor de aplicar apply2 con una acción concreta al estado del padre. Concretamente, al hacer actFORWARD a una casilla con bikini o zapatillas, en apply2 se actualizan las variables pertinentes. Si la acción no es actFORWARD y se estaba en una casilla sin ninguno de estos dos items, las variables booleanas se mantienen con los valores de las del padre.

Nivel 3-A* para el agente sonámbulo

- a ¿Qué diferencia este algoritmo del de Dijkstra que tuviste que implementar en el nivel anterior? (enumera los cambios y describe brevemente cada uno de ellos y que han implicado en la implementación)

La diferencia principal es que aparte de tener un coste asociados a los nodos, también se tiene una heurística para ellos. Voy a enumerar los cambios respecto al nivel 2:

1) Ahora stateN3 es como stateN1 pero añadiendo 4 variables booleanas que nos indican si el jugador o el sonámbulo tienen el bikini o las zapatillas. Además, estas también se tienen en cuenta en los operadores == y < (recordar que se trata de una comparación lexicográfica, como si de palabras se tratase).

2) Ahora nodeN3, al igual que nodeN2, se construye sobre su respectivo state (stateN3) y posee también su operador < (que se usará para frontier (cola con prioridad)) donde ahora lo que se compara es el coste + heurística. Como en el nivel 2 lo que usaba como coste era la batería, tuve que realizar una pequeña modificación para que esta comparación tuviese sentido (porque si no, la heurística la comparaba con < y batería con >).

3) Creé una función llamada CosteAccion3, que es igual que CosteAccion del nivel 2 pero añadiendo las acciones del sonámbulo, las cuales antes no se consideraban porque lo que había que llevar era al jugador.

4) En el algoritmo A*, la estructura es prácticamente la misma a la del nivel 2, solo que generando también los hijos correspondientes a las acciones del sonámbulo cuando este se ve (VeoSonambulo) y actualizando la heurística de los nodos solo cuando se mueve el sonámbulo. Esto es porque tal y como está definida la heurística, (distancia máxima en filas y columnas del sonámbulo al objetivo), está claro que si se mueve el jugador, o se realiza cualquier acción de giro, la heurística sigue siendo la misma.

b Describe la heurística utilizada para resolver el problema

La heurística empleada es la distancia del máximo del sonámbulo al objetivo. Consiste en calcular la diferencia en valor absoluto entre la fila del sonámbulo y la del objetivo; y calcular la diferencia en valor absoluto entre la columna del sonámbulo y la del objetivo. Una vez calculadas, se le asigna a la heurística del nodo el máximo de estas dos diferencias.

Nivel 4-Reto (Max. Puntuación en misiones)

a Haz una descripción general de tu estrategia general con la que has abordado este nivel. Indica bajo qué criterios es el jugador o el sonámbulo el que va al objetivo. Explica brevemente las razones de esos criterios.

b ¿Qué algoritmo o algoritmos de búsqueda usas en el nivel 4? Explica brevemente la razón de tu elección.

c ¿Bajo qué condiciones replanifica tu agente?

d Explica el valor que le has dado a la casilla desconocida en la construcción de planes cuando el mapa contiene casillas aún sin conocer. Justifica ese valor.

e ¿Has tenido en cuenta la recarga de batería? En caso afirmativo, describe la política usada por tu agente para proceder a recargar.

f ¿Has tenido en cuenta la existencia de aldeanos y lobos para definir el comportamiento del agente? En caso afirmativo, describe en qué sentido los has tenido en cuenta.

g Añade aquí todas los comentarios que desees sobre el trabajo que has desarrollado sobre este nivel, qué consideras que son importantes para evaluar el grado en el que te has implicado en la práctica y que no se puede deducir de la contestación a las preguntas anteriores.

Comentario final

Consigna aquí cualquier tema que creas que es de relevancia para la evaluación de tu práctica o que quieras hacer saber al profesor.