

INTELIGENCIA ARTIFICIAL

CURSO 2022-23

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

| | | | |
|--------------------|-----------------------|-----------------|-----|
| APELLIDOS Y NOMBRE | Cribillés Pérez María | | |
| GRUPO TEORÍA | GDIIM | GRUPO PRÁCTICAS | AD3 |

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):

Nivel 0, nivel 1, nivel 2, nivel 3 y nivel 4

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)

He usado herencia entre estados, por tanto el stateN1 hereda de stateN0. Al stateN1 se sobrecarga los operadores == y < ya que al considerar también al sonámbulo hay que comparar su fila, columna y orientación.

En la función en sí, usamos nodosN1, la solución es cuando el sonámbulo llegue al destino (en vez del jugador).

En el del n0 solo se hacían 3 tipos de hijos (las acciones del jugador), mientras que en el n1 hacemos 6 (posibles acciones del jugador y sonámbulo).

En el apply también consideramos las posibles acciones del sonámbulo y solo realiza las acciones del sonámbulo si el jugador ve al sonámbulo.

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)

He realizado un a* en vez de coste uniforme como se propone. Es lo mismo solo que le he incluido la distancia manhattan para el jugador como heurística. Como solo uso el jugador, no creo hijos del sonámbulo. Además, en cada momento le sumo el coste de la casilla de la que salgo al coste acumulado anterior.

Compruebo la solución una vez que entra a cerrados porque podría haber una posible mejora, no como en anchura que una vez que entra en abiertos y es solución ya paramos. En anchura es así porque los abiertos son una cola entonces salen en el mismo orden en el que entran.

Además, usamos una cola con prioridad en los abiertos para que se ordenen cuando se meten

según prioridad. Para ello hemos hecho un nuevo stateN2 añadiendo los objetos especiales del jugador y sobrecargando los operadores con estas nuevas opciones.

- (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, cuando se generaba un hijo y se mete en la secuencia llamo a la función actualizarObjetos, la cual si la casilla donde está mirando es bikini, suelta las zapatillas y actualiza a true el bikini en el estado. Lo mismo para zapatillas. Hago comprobaciones que realmente no hacen falta, ya que da igual si tienes o no el otro objeto especial, lo pones a false si te encuentras el que no tenías y ya.

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)

He cambiado la heurística porque se considera al sonámbulo también que puede realizar los 8 giros, mientras que antes el jugador solo podía realizar giros de 90 grados.
Como es el sonámbulo el que tiene que llegar al objetivo, el algoritmo de búsqueda parará cuando éste llegue al destino final.
Se consideran sus posibles acciones solo si el jugador le ve.
He añadido al stateN3 las variables de zapatillas_son y bikini_son. Compruebo cada vez que genero hijos y los meto en secuencia si está en una casilla de objeto especial y lo actualizo a su correspondiente jugador. También si es la primera casilla.
Este algoritmo al generar tantas opciones, ya que tenemos los 6 hijos (3 de cada jugador) tarda mucho en ejecutarse y en pensar, por lo que no lo he usado para implementar el nivel 4.

- (b) Describe la heurística utilizada para resolver el problema

He usado la distancia de Chebychev o la del máximo.
Se calcula el máximo entre el valor absoluto de las diferencias entre filas y columnas. Es decir:
$$\text{distanciaMaximo}(x,y) = \max(|x1-x2|, |y1-y2|)$$

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.

Los jugadores realizan planes independientes intentando que sea el sonámbulo el que llegue al objetivo y el jugador que en todo momento vea al sonámbulo y su siguiente casilla para que así pueda ver si se choca o no.
Si por alguna razón, o bien el sonámbulo no llega a la solución o bien el jugador no llega a ver al sonámbulo, es el jugador el que va a por los objetivos.

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

Uso un a^* del n^2 para cada jugador, es decir, realizan planes independientes. El jugador avanzará hasta ver al sonámbulo y su siguiente casilla, mientras que el sonámbulo va al objetivo. Creo que es mejor opción que plantear un a^* estrella que cree hijos del jugador y del sonámbulo ya que lo intenté, pero tardaba mucho en pensar y acabé entendiendo más lo que estaba haciendo con la opción realizada.

(c) ¿Bajo qué condiciones replanifica tu agente?

Tanto para sonámbulo como para jugador: si se encuentra muro o precipicio, si tiene poca batería va a por ella, si tiene delante bosque sin zapatillas o si tiene delante agua sin bikini.

(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.

Le he dado coste 1 y no he aumentado la heurística, es decir, que es la distancia tal cual. Creo que es mejor ya que así va a preferir recorrer mapa desconocido a si por ejemplo hay agua o bosque (mayor coste sin objetos especiales). Esto en algunos mapas donde hay mucho bosque o agua es perjudicial, pero solo pasa en los mapas como paldea.

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

Sí. Cuando tenía menos de 1000 batería y también según los ciclos que queden, busca si ha visto una recarga en el mapa y va a por ella. Una vez allí recarga unas 130 veces si no se encuentra ningún lobo delante que sale huyendo.

(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.

Sí, si me encontraba en la casilla de delante un aldeano o un lobo e iba a avanzar, he hecho actIDLE hasta que se iba. Cuando se iba volvía a replanificar si no tenía plan.

(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.

Tenía planteado y programado también que cuando quedasen pocos ciclos hiciese anchura porque lo que queremos es que llegue al objetivo en los menos ciclos posibles. Sin embargo, tardaba mucho en pensar y por miedo al timeout lo dejé comentado. Por la parte de los empujones de los lobos, solo hago el WHEREIS cuando creo que me han empujado. Podría haberlo mejorado planteando ver lo que veía antes y lo que veo ahora y ver si coincidían pero ya no me dio tiempo. Creo que hubiese sido una mejora importante ya que ahorraría bastante batería. Hay veces que cuando el sonámbulo se le escapa del campo de visión del jugador y va a por el,

creo que no tiene mucho en cuenta si no tiene objetos especiales y hay agua o bosque. Esto también sería un punto a mejorar porque creía haberlo hecho pero me di cuenta que en justo ese caso debería de volver a buscar si puede volver a ver al sonámbulo sin tener que llegar a gastar mucha batería.

Otra mejora sería que el jugador busca con a* una casilla para ver al sonámbulo y su siguiente casilla, pero de las posibles soluciones que hay, yo voy a la primera. Podría ver cual es la de menos coste e ir a esa, ya que al salir me costará menos. La menos recomendada sería la que justo esa casilla sea bosque/agua sin objetos especiales .

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.

Todo bien. Podría haberle hecho alguna mejora más que he dejado planteada en los apartados anteriores (en el anterior sobre todo).