

INTELIGENCIA ARTIFICIAL

CURSO 2023-24

PRACTICA 1: Repertorio de preguntas para la autoevaluación de la práctica 1.

APELLIDOS Y NOMBRE

Rodríguez Faya, Ángel

GRUPO TEORÍA D

GRUPO PRÁCTICAS

D2

Instrucciones iniciales

En este formulario se encontrarán preguntas que tienen que ver con (a) descripciones en lenguaje natural del comportamiento implementado en tu agente o (b) con resultados sobre ejecuciones concretas del software desarrollado por los estudiantes para problemas muy concretos.

En relación a los resultados sobre ejecuciones concretas, estas se expresarán usando la versión de invocación en línea de comandos cuya sintaxis se puede consultar en el guión de la práctica. Para ello, toma los nuevos mapas (*mapa50_eval2324.map*, *mapa75_eval2324.map* y *mapa100_eval2324.map*) que se adjuntan con la autoevaluación y cópialos en la carpeta **mapas** donde se encuentre tu software.

Antes de empezar, ten en cuenta las siguientes consideraciones:

- **Asegúrate de tener la versión más reciente del software descargada.** Para ello, si utilizas el repositorio de GitHub, puedes hacer git pull upstream main tras haber seguido los pasos del README. Si no, descárgate el zip con la carpeta de la práctica y copia dentro tus archivos jugador.cpp y jugador.hpp.
- **Si consideras que en alguna de las ejecuciones los lobos u otros elementos te han perjudicado considerablemente en el resultado añádelo como comentario junto con el resultado.** También puedes proponer una semilla alternativa en tal caso en los comentarios.
- **El software corrige la orientación automáticamente, aunque se le pase un valor que no sea norte en los niveles que solo admiten dicha orientación.** No es necesario hacer ningún cambio en los comandos que se os piden en ningún nivel.

Poner en los recuadros la información que se solicita.

(a) *Describe de una manera simple, breve y concisa (usando lenguaje natural) como has definido la forma en la que tu agente se mueve.*

El movimiento de mi agente es el siguiente:

En primer lugar, se recorre todo el vector de sensores. terreno del agente, y hay una serie de casillas en un orden descendente, siendo la primera la de posicionamiento y la última la de las zapatillas, en función de donde se encuentren (si a la izquierda en el centro o a la derecha del agente) se elegirá la siguiente acción (**Función siguienteAcción**). Después, se evalúa si es necesario que el agente siga una serie de reglas, que he definido para que pueda salir cuando esté encerrado. Por ejemplo, si está encerrado y paralelo al muro o precipicio y ve un hueco, gira y se mete por dicho hueco. Finalmente encontramos si el agente ha sido reiniciado en una casilla a la que he definido como “bloqueante” y si el agente percibe un lobo en un sus sensores. En esta última, si lo percibe, girará a un lado o a otro pero no se sabe a priori ya que se decide de forma aleatoria (**Función giroRandom**). En el “reinicio bloqueante”, que también sirve para la primera iteración en la que puede ocurrir lo mismo, se da debido a que yo he puesto como casilla a las que se pueden avanzar el agua y el bosque (pero no puedes avanzar si no tiene bikini o zapatillas, respectivamente), entre otras. Por lo que tras ser reiniciado o en la primera iteración, si estamos en una casilla de bosque o agua, y no llevamos el bikini o las zapatillas, podrá realizar 10 iteraciones en las que avanzará si puede, o si hay algún muro o precipicio girará, todo ello lo hará aunque le cueste más batería. Una vez finalizadas las 10 iteraciones, volverá a la normalidad.

Cabe destacar que:

- Los giros son aleatorios, por lo que no se sabe si girará a la derecha o a la izquierda.
- Si no se decide ninguna acción en concreto o ha habido alguna controversia, no se hará nada, **actIDLE**.

(b) *¿Tu agente va de forma activa hacia los objetos cuando estos aparecen en su sensor de visión? En caso afirmativo, describe la forma en que se implementa ese comportamiento activo.*

No. Como ya he indicado en el apartado anterior, cuando los objetos aparecen en el sensor de visión de mi agente, este girará o avanzará dependiendo de donde se encuentren dichos objetos, si a la derecha, izquierda o al centro, pero no va directo a ellos.

(c) *¿Influye en el comportamiento que has definido el hecho de tener o no el bikini o las zapatillas? En caso afirmativo describe la forma en la que influye.*

Si, como he explicado en el apartado a), cuando no se tienen el bikini o las zapatillas, no podrá avanzar sobre las casillas de agua o de bosque hasta que el agente se las ponga. Existen casos especiales, como lo son un reinicio o en la primera iteración cuando el agente aparece en el mapa, que se puede dar el caso en

el que el agente aparezca en las casillas de agua o bosque, y tendrá que avanzar por ellas aunque no lleve los complementos puestos.

(d) *¿Has tenido en cuenta en el comportamiento la existencia de casillas que permiten la recarga de batería? En caso afirmativo describe cómo lo has tenido en cuenta.*

Si, cuando el agente se posiciona en una casilla de Recarga y tiene menos de cuatro mil puntos de batería, se recargará 10 puntos de batería por cada iteración, hasta que sea mayor o igual a cuatro mil puntos, en ese caso abandonará la casilla. Para ello he diseñado la función **recargarPilas** y las variables de estado **necesita_recargar** y **recargando_pilas**. Además, si le queda poca vida (menos de 100 iteraciones para terminar) no se recargará la batería.

(e) *¿Has definido alguna estrategia para intentar eludir las colisiones con los aldeanos y los lobos?*

Si, cuando encuentre un lobo o aldeano en su sensor de visión, girará, intentando evitarlo.

(f) *¿Has incluido comportamientos que son específicos para los niveles 2 y 3? Describe los comportamientos y brevemente las razones que te impulsaron a incluirlos.*

Respecto a los agentes que pueden aparecer estos dos niveles(el 2 y el 3), sólo he implementado lo dicho en el apartado anterior. Además, he implementado cómo se actualizarán los sensores una vez se posicionan en la casilla de posicionamiento, codificada con el carácter 'G'. También destacar que la función de pintar en el mapa resultado lleva unos condicionales en los que si estamos en el nivel 3, no pintará los sensores que fallan en ese nivel (6, 11, 12 y 13). Por último, he implementado el tema del reinicio en caso de que un lobo se coma al agente.

(g) *¿Has implementado algún comportamiento para llevar a cabo la acción actRUN en lugar de actWALK en determinadas situaciones? En caso afirmativo, en el nivel 3, ¿cómo has gestionado la ausencia de sensores de visión (posiciones 6, 11, 12 y 13)?*

No, no he llegado a implementar actRUN. Solamente en caso de utilizarlo, como quedarían las variables de estado.

(h) *Describe cuáles son los puntos fuertes de tu agente.*

Considero que mi agente tiene como puntos fuertes:

- Pinta el mapa tal y como es, no mueve casillas ni pinta lo que no es.
 - Descubre gran parte del mapa.
 - No se mete en casillas como el agua o el bosque en las que puede gastar mucha batería sin llevar los complementos.
 - No da problemas al reiniciar o empezar el juego en casillas bloqueantes.
 - Tiene en cuenta todos los aspectos del juego, así como las restricciones que hay en cada nivel.
-

(i) *Describe cuáles son los puntos débiles de tu agente.*

Considero y admito que mi agente tiene como puntos débiles:

- Se puede quedar atascado con facilidad, o gastar varias iteraciones moviéndose en el mismo sitio, por ejemplo, estar un poco entre esquinas moviéndose todo el rato hacia arriba y hacia abajo.
 - No tiene un plan o un movimiento sofisticado para llegar a ciertas casillas, a veces puede pasar al lado de una casilla de posicionamiento y no estar bien situado, y en vez de pararse en ella, sigue avanzando.
-

(j) *Incluye aquí todos los comentarios que desees expresar sobre la práctica que no hayas descrito en las preguntas anteriores.*

(k) *Ejecución 1: Ejecuta el siguiente comando en un terminal*

`./practica1SG mapas/mapa50_eval2324.map 1 n 30 6 4`

para los 5 valores de **n**, desde 0 hasta 3 y coloca los resultados de porcentaje de mapa descubierto con dos decimales en la siguiente tabla. Si la ejecución da un error y no termina dando un resultado, pon "core" en la casilla de la tabla correspondiente.

n = 0	n = 1	n = 2	n = 3
86.72	84.16	65.24	55.8

(l) *Ejecución 2: Ejecuta el siguiente comando en un terminal*

`./practica1SG mapas/mapa75_eval.map 1 n 47 6 2`

para los 5 valores de **n**, desde 0 hasta 4 y coloca los resultados de porcentaje de mapa descubierto con dos decimales en la siguiente tabla. Si la ejecución da un error y no termina dando un resultado, pon “core” en la casilla de la tabla correspondiente.

n = 0	n = 1	n =2	n =3
46.93	51.00	69.94	90.37

(m)Ejecución 3: Ejecuta el siguiente comando en un terminal

`./practica1SG mapas/mapa100_eval.map 1 n 57 95 6`

para los 5 valores de **n**, desde 0 hasta 4 y coloca los resultados de porcentaje de mapa descubierto con dos decimales en la siguiente tabla. Si la ejecución da un error y no termina dando un resultado, pon “core” en la casilla de la tabla correspondiente.

n = 0	n = 1	n =2	n =3
52.57	66.97	45.96	63.68