INTELIGENCIA ARTIFICIAL

CURSO 2022-23

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

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

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 guion de la práctica. Para ello, toma los nuevos mapas (***mapa30_eval.map***, ***mapa50_eval.map*** y ***mapa75_eval.map***) que se adjuntan con la autoevaluación y cópialos en la carpeta ***mapas*** donde se encuentre tu software.

Consideraciones importantes:

- Antes de empezar a rellenar el cuestionario, actualiza el código de la práctica con los cambios más recientes. Recuerda que puedes hacerlo o bien realizando **git pull upstream main** si has seguido las instrucciones para enlazar el repositorio con el de la asignatura, o bien descargando desde el enlace de GitHub el zip correspondiente, y sustituyendo los ficheros **jugador.cpp** o **jugador.hpp** por los vuestros.

- Si en alguna ejecución consideras que tu agente se ha visto perjudicado puedes añadirlo a los comentarios finales o comentarlo al lado de la ejecución afectada, y proponer una semilla diferente o posición cercana que tendremos en cuenta.

Poner en los recuadros la información que se solicita.

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

La forma en la que se mueve mi agente es por decirlo así “puntos de gravedad”. Existe un método que calcula cuál de las tres casillas 1, 2 y 3 es la más prioritaria (una casilla es más prioritaria cuanto menos frecuencia de visitas tenga. Si el agente pasa por una casilla, se le suma 5 a su frecuencia en una matriz auxiliar llamada contador (o contador2 si está desorientado) y se le suma 1 a las que se ven). Por lo tanto, cada 250 instantes, se calcula cuál es la casilla objetivo del mapa, es decir, la casilla desconocida más cercana al agente. Una vez calculada, se guarda en una variable pair objetivo y se cambian todas las frecuencias del mapa, poniendo a 0 la frecuencia del objetivo y las demás frecuencias se calculan como la

distancia en filas y columnas de cada casilla a la casilla objetivo. De esta forma, es como si se hiciese un “punto de gravedad”, pues con el método de ver cuál es más prioritaria de la 1, 2 y 3, el agente va cayendo a la zona que me interesa. Una vez visto el objetivo (la casilla deja de ser desconocida), se reestablecen las frecuencias hasta que pasan 250 instantes y vuelta a empezar.

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.*

Sí. Cuando avista unas zapatillas, un bikini, una casilla de recarga o de posicionamiento, lo que se hace es dirigirlo hacia esa zona. Entonces, si la casilla avistada es la 1, 4 o 9, debe hacer un giro pequeño a la izquierda. Si es la 2, 5, 6 , 7, 10, 11, 12, 13 o 14, avanza hacia delante. Si es la 3, 8 o 15, hace un giro pequeño hacia la derecha para orientarse bien para el siguiente instante.

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.*

Sí que influye. Tengo dos variables llamadas zapatillas y bikini en la clase de ComportamientoJugador. Comienzan a false hasta que el agente logra hacerse con estos ítems. Cuando están a false, en caso de empezar en agua o bosque por un reinicio, busca por sus alrededores alguna casilla que no sea agua ni bosque y se dirige hacia ella de la misma forma en la que se ha descrito el comportamiento para ir a una casilla especial en b). Le permito pisar el bosque cuando la batería no sea baja aunque no tengo zapatillas, poniéndole un límite de 2000, mientras que el agua directamente la considero como casillas prohibidas una vez sale de ella. Para que no se meta mucho en bosque mientras que la batería está por encima de 2000, tengo una función llamada NoBosqueCerca que se usa cuando la casilla en la que está es bosque o las que tiene delante (1, 2 y 3) lo son, para que localice una casilla que no sea bosque dentro de los sensores de su vista y se dirija ahí para no salir del bosque.

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.*

Sí. De hecho voy guardando las posiciones de las recargas de batería en un set de pair llamado recargas para cuando está bien orientado; y recargas2 cuando no está bien orientado. He escogido un set como almacenamiento porque así no me tengo que preocupar de si va a insertar la misma casilla de recarga, pues el set gestiona que no haya repeticiones. Así, cuando

está bajo de batería (< 1500) y conoce alguna casilla de recarga, aplico el método de gravedad explicado en el apartado a) para que acabe cayendo a la casilla de recarga y se quede ahí hasta que llegue a 3000 de batería.

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

Sí. Lo único que he hecho ha sido implementar al final del código que si la acción que se ha determinado hacer sea actFORWARD y resulta que hay un aldeano o un lobo, se espere haciendo actIDLE.

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.*

Nada distinto de lo expuesto en el apartado anterior (e). También en cada turno compruebo si ha habido un reinicio y en caso de haberlo, actualizo la fila, columna, orientación, borro el bikini y las zapatillas, etc.

g *Describe cuáles son los puntos fuertes de tu agente.*

Los puntos fuertes de mi agente son:

1. Que cuando ve una casilla con algo que le hace falta (recarga, posicionamiento, zapatillas o bikini) va hacia ella sin falta salvo que esté sorteando un muro.
2. La modificación de frecuencias según lo que me interesa, pues con el método descrito según la distancia de la casilla a la que quiero ir, logro dirigir al agente hacia donde necesito.

h *Describe cuáles son los puntos débiles de tu agente.*

Diría que el gran punto débil es en caso de reiniciarse en un gran bosque o gran charco, pues lo tengo programado para que busque en el punto en el que está y en caso de que no encuentre casillas que no sean bosque o agua, se gire y vuelva a probar con la nueva vista. Sin embargo, si se gire hacia donde se gire solo hay agua o bosque, no saldría de ahí pues no le he indicado que avance cuando no encuentre nada. Este comportamiento no lo he programado ya que tras

múltiples pruebas y ejecuciones, nunca se ha reiniciado en un punto tan concreto en el que pase eso a no ser que lo coloque yo a posta.

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

j *Ejecución 1: Ejecuta el siguiente comando en un terminal*

./practica1SG mapas/mapa30_eval.map 1 0 5 20 7

para el nivel n = 0 y

./practica1SG mapas/mapa30_eval.map 1 n 5 20 0

para los 3 valores de n, desde 1 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	n = 4
96.4444	100	100	98.6667	

k *Ejecución 2: Ejecuta el siguiente comando en un terminal*

./practica1SG mapas/mapa50_eval.map 1 0 23 41 5

para el nivel n = 0 y

./practica1SG mapas/ mapa50_eval.map 1 n 23 41 0

para los 3 valores de n, desde 1 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	n = 4
90.64	98.72	99.28	99.96	

- *Ejecución 3: Ejecuta el siguiente comando en un terminal*

  ./practica1SG mapas/mapa75_eval.map 1 0 12 7 4

  para el nivel n = 0 y

  ./practica1SG mapas/ mapa75_eval.map 1 n 12 7 0

  para los 3 valores de n, desde 1 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	n = 4
76.7822	95.8044	95.8756	66.8089	

Aquí me gustaría añadir que en n = 3, la ejecución no es muy buena porque mi agente resulta tener mala suerte. Cuando todavía tiene más de 1500 y tiene casillas de recarga registradas, lo mata un lobo y se reinicia justo en el agua al lado de un muro. Como se topa con el muro, primero sale de él y se come el agua, perdiendo mucha batería y entrando en un estado crítico de batería. Cuando sale del agua y el muro, está desorientado así que no puede dirigirse a las casillas de recarga registradas. De hecho, cuando se ubica, se pone en dirección a la casilla de recarga pero no le da la batería para llegar. He probado el nivel 3 en una casilla aleatoria, la primera que se me ha ocurrido. En este caso la 20 20 con orientación norte me ha dado buen resultado porque no le ha pasado esa desgracia, obteniendo un porcentaje del 85.0311. A parte de proponer esa casilla de inicio, he preguntado hoy en clase de prácticas y me han dicho que lo que tengo que proponer para ese caso son otras semillas. He estado probando y por ejemplo, las semillas 3, 4 y 5 dan porcentajes mucho mejores que 66 (lo de las semillas lo he probado con gráfica porque según me he enterado, sin gráfica siempre se ejecuta con semilla 1).