

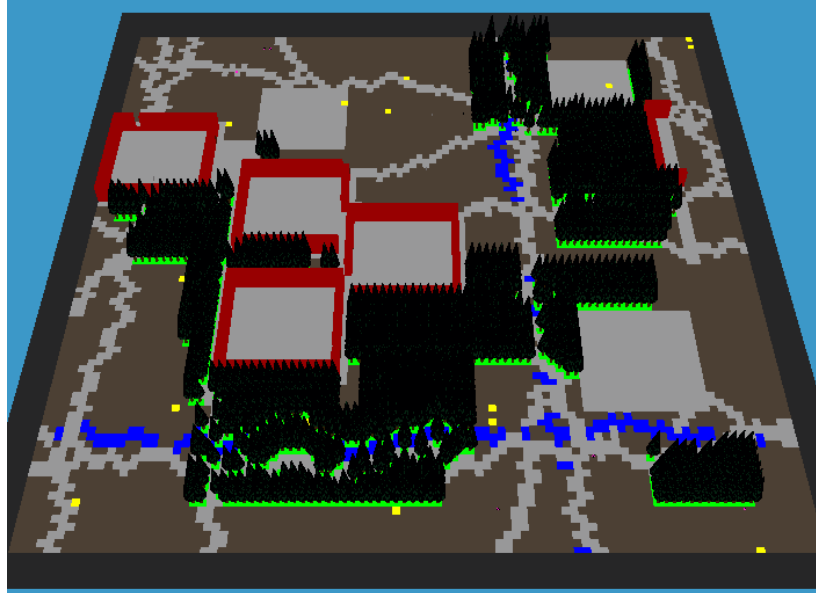
INTELIGENCIA ARTIFICIAL

E.T.S. de Ingenierías Informática y de
Telecomunicación

Ejemplos prácticos: Práctica 2

Agentes Reactivos/Deliberativos

(Los extraños mundos de BelKan)



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA
ARTIFICIAL

UNIVERSIDAD DE GRANADA

Curso 2021-2022

1. Introducción

Este documento se aporta, de forma complementaria al material disponible de la Práctica 2, con el fin de aclarar el comportamiento esperado en cada uno de los niveles descritos en el guión mediante ejemplos concretos. Dado que el Nivel 0 se implementa siguiendo el “Tutorial: Práctica 2”, se mostrarán ejemplos prácticos a partir del Nivel 1.

2. Nivel 1

En el Nivel 1 se pide encontrar el camino óptimo en cuanto a número de acciones. Por tanto, el comportamiento que se espera tras implementar la búsqueda en anchura es que el agente encuentre el camino que necesite un menor número de acciones independientemente del terreno que se encuentre (siempre que sea transitable). Esto es:

- Si no se encuentran obstáculos entre la posición del agente y el objetivo, el camino solución será lo más recto posible minimizando el número de giros, que contabilizan como acciones.

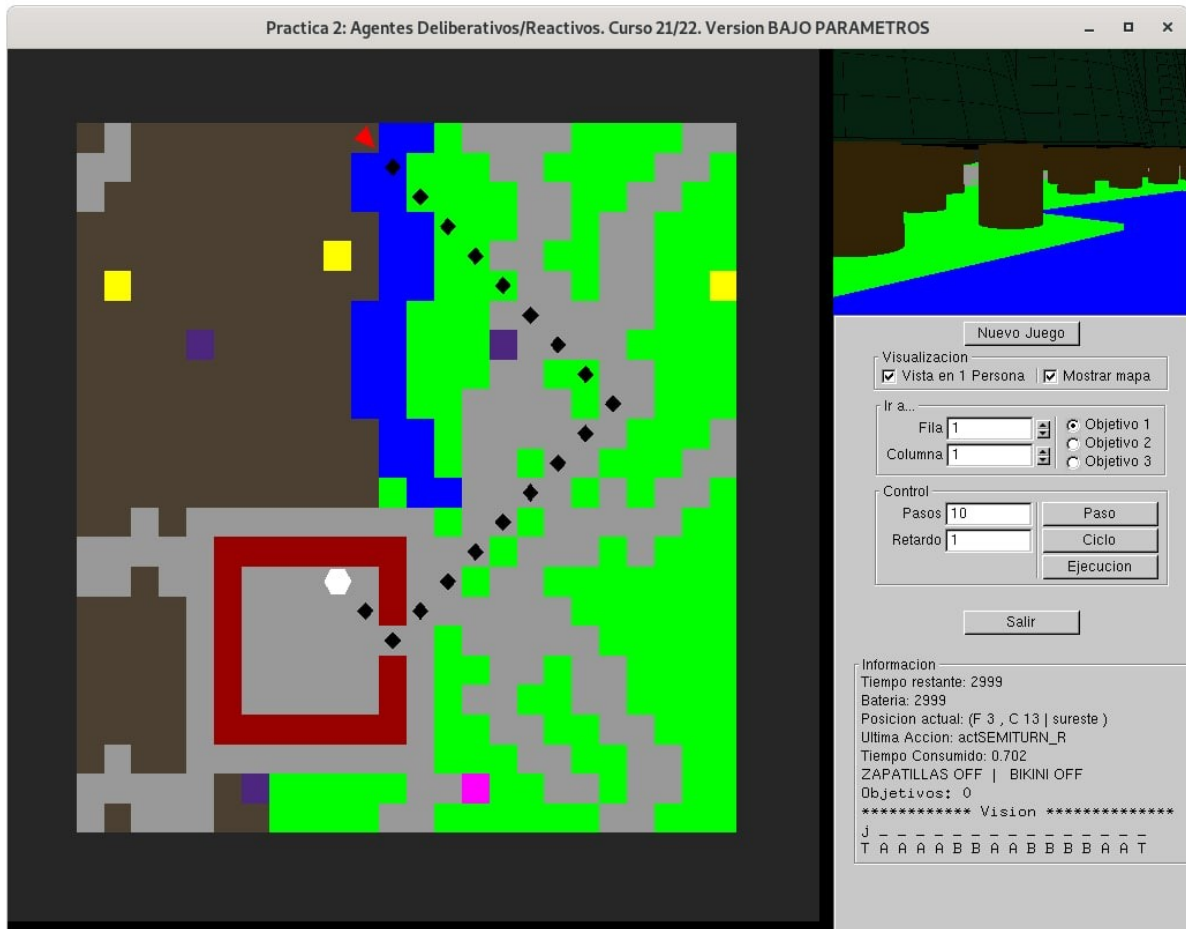
Ejemplo 1: `./practica2 mapas/mapa30.map 1 1 3 13 2 13 13`



Instantes de simulacion no consumidos: 2988
Tiempo Consumido: 0.066711
Nivel Final de Bateria: 1790
Plan: D A A A A A A A A A A (longitud 11)

- En el caso de que se encuentre algún obstáculo no transitable (por ejemplo, muros), el agente deberá esquivarlo por el camino más corto y que minimice el número de giros, de nuevo, de forma independiente al coste de transitar cada terreno.

Ejemplo 2: `./practica2 mapas/mapa30.map 1 1 3 13 2 18 12`



Instantes de simulacion no consumidos: 2977
 Tiempo Consumido: 0.703101
 Nivel Final de Bateria: 2283
 Plan: d A A A A A A A A A D A A A A A A A A D A A (longitud 22)

Es importante tener en cuenta que la búsqueda en anchura **debe ser eficiente**. La búsqueda en anchura necesita mucha memoria para funcionar correctamente, por lo que la implementación debe ser lo más eficiente posible. En esta búsqueda los abiertos se sacan en el mismo orden en el que entran, por lo que una implementación adecuada debería poder parar en cuanto se genera la solución, y no esperar hasta que esta se extraiga de abiertos. También es importante que no se visiten nodos repetidos y solo se expandan los nodos la primera vez que se visitan. Esto ha de tenerse en cuenta al desarrollar el algoritmo para la práctica.

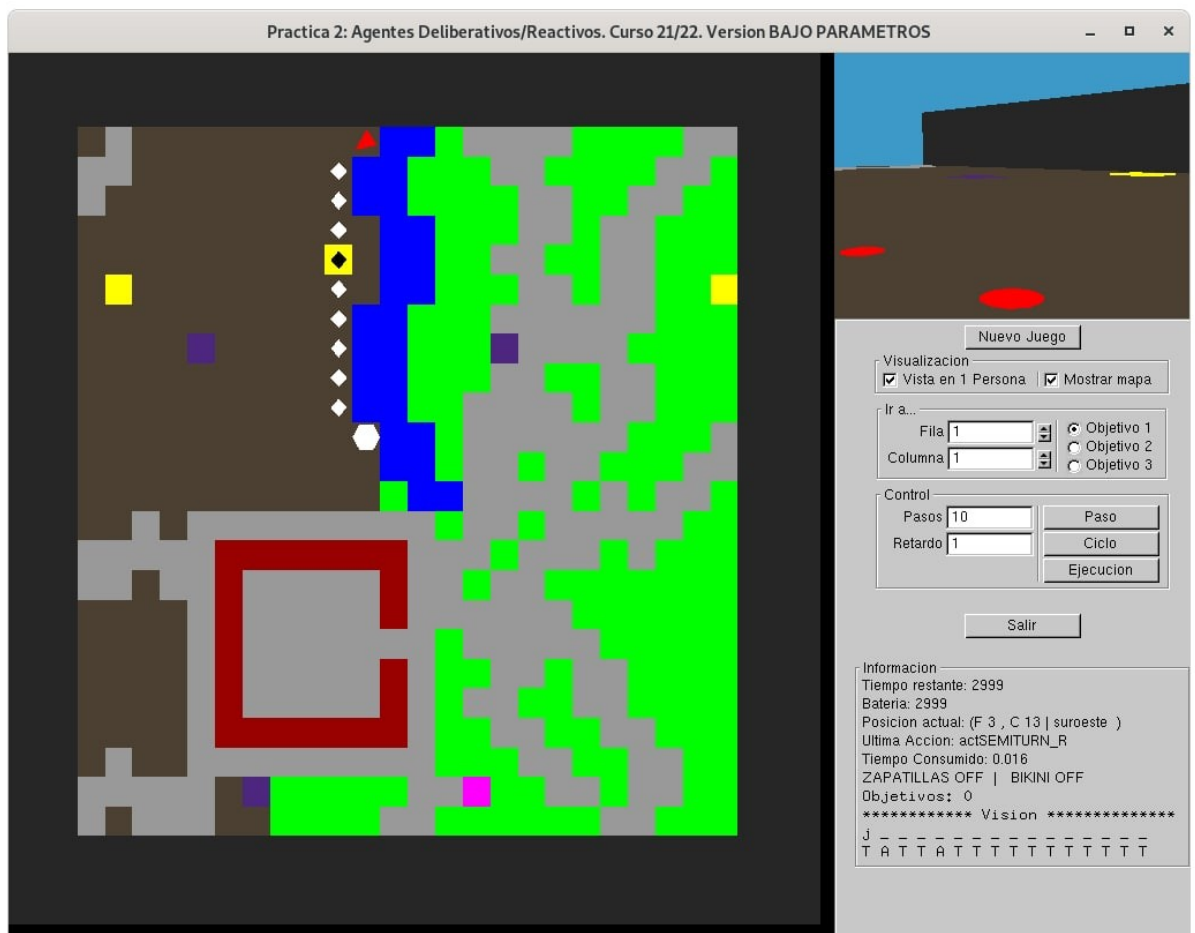
3. Nivel 2

La principal limitación del Nivel 1 es que se encuentra el camino óptimo en cuanto a número de acciones, lo cual no tiene por qué corresponderse con el mínimo consumo de batería dado que en el Mundo de Belkan nos encontramos con diferentes tipos de terreno que conllevan consumo diferente de batería. Esto puede producir que se consuma mucha más batería de la necesaria, llegando incluso a agotarse.

En este nivel, se pide encontrar el camino que lleve al agente al objetivo que consuma menos batería. Para ello, se pide implementar una búsqueda por coste uniforme o utilizando el algoritmo A*. Los resultados esperados serán, por regla general, más largos que los del Nivel 1 en cuanto a número de acciones pero optimizarán el consumo de batería de diferentes maneras: eligiendo caminos menos costosos (como suelo pedregoso) o utilizando el bikini o las zapatillas para minimizar el coste de transitar aquellos terrenos más costosos.

En los ejemplos 3 y 4, el agente parte del mismo origen y tiene que llegar al mismo objetivo que en los ejemplos 1 y 2, respectivamente. Sin embargo, en este nivel logramos que se incremente notablemente el nivel final de batería pasando de 1790 a 2978 y de 2283 a 2964 respectivamente, aunque se incrementen el número de acciones al realizar más giros.

Ejemplo 3: ./practica2 mapas/mapa30.map 1 2 3 13 4 13 13



Instantes de simulacion no consumidos: 2986

Tiempo Consumido: 0.016216

Nivel Final de Bateria: 2978

Plan: d A i A A A A A A A i A (longitud 13)

Ejemplo 4: ./practica2 mapas/mapa30.map 1 2 3 13 4 18 12



Instantes de simulacion no consumidos: 2975

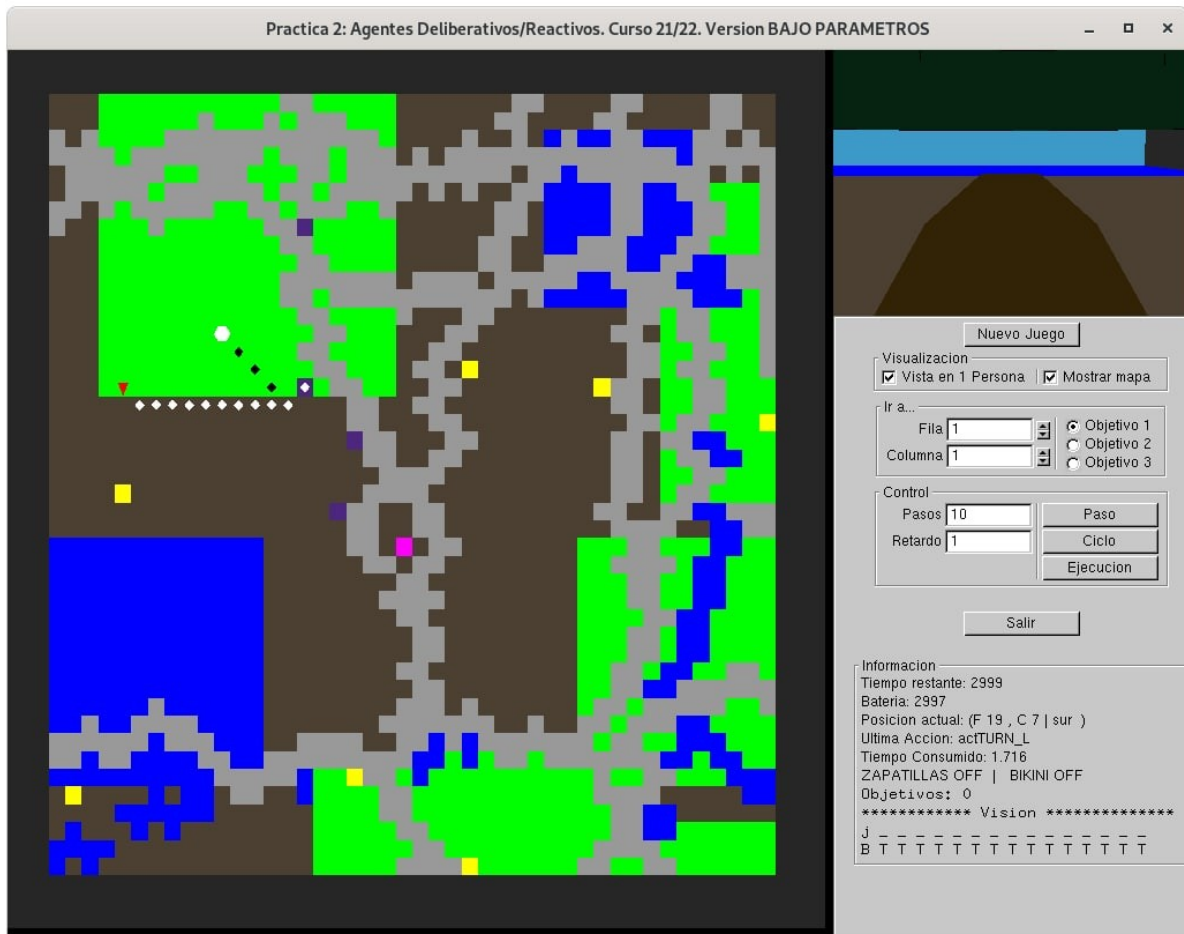
Tiempo Consumido: 0.11167

Nivel Final de Bateria: 2964

Plan: d A i A A A A A A A A A i A A A A D A A D A A (longitud 24)

En el ejemplo 5, vemos como el agente da un rodeo antes de entrar al bosque para conseguir las zapatillas, lo cual disminuirá el consumo de batería de andar por el bosque, consiguiendo un resultado óptimo en cuanto a consumo de batería.

Ejemplo 5: ./practica2 mapas/mapa50.map 1 2 19 7 6 16 13



Instantes de simulacion no consumidos: 2975

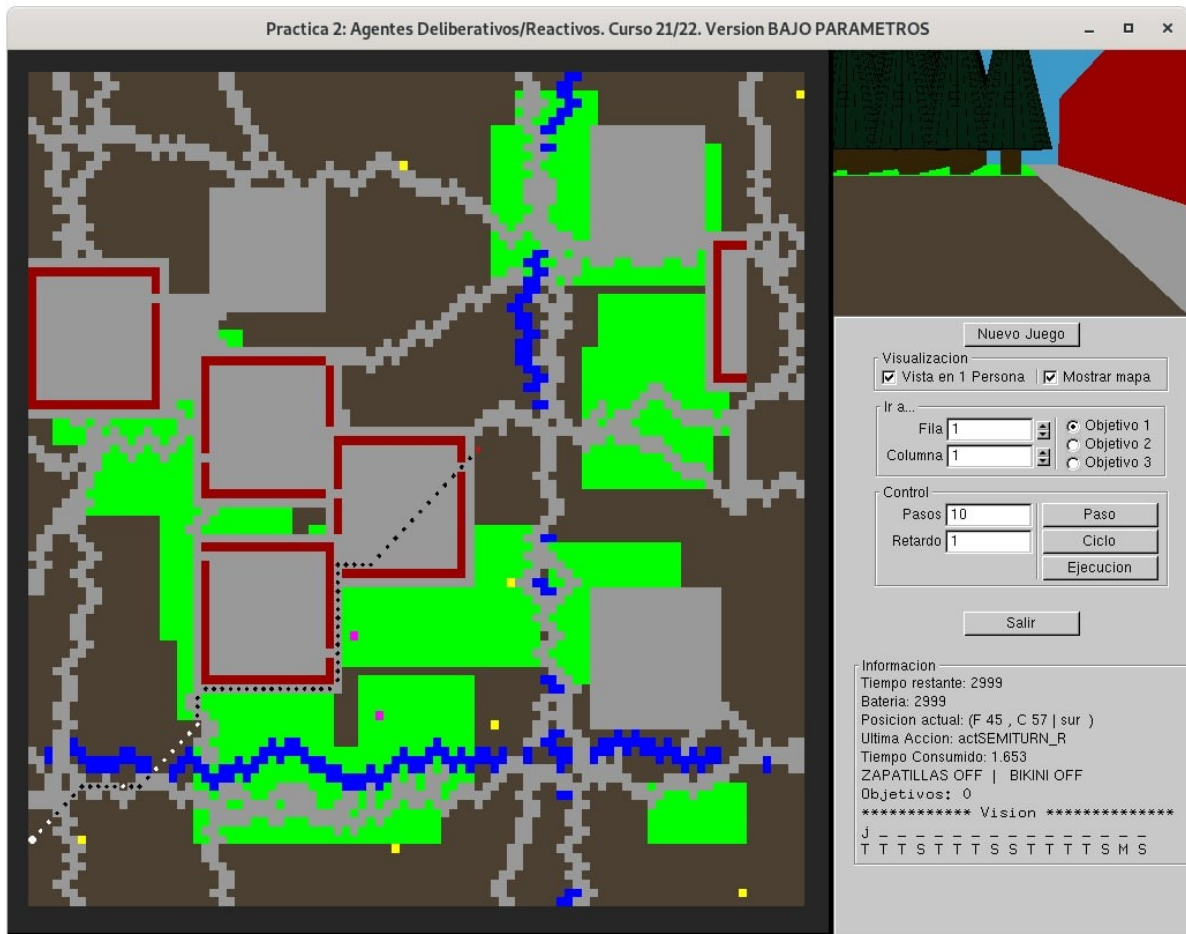
Tiempo Consumido: 1.71726

Nivel Final de Bateria: 2821

Plan: I i A i A A A A A A A A A i A D D A d d A A A A (longitud 24)

En caso de optar por hacer el A*, es importante usar una heurística apropiada que sea admisible. Si no es admisible, el camino que puede encontrar el A* al objetivo podría no ser el de coste óptimo. En los ejemplos 6 y 7 vemos este comportamiento. En el ejemplo 6 aplicamos el A* con una heurística admisible, mientras que en el ejemplo 7 lo aplicamos con una heurística no admisible.

Ejemplo 6: ./practica2 mapas/mapa100.map 1 2 45 57 3 89 3



Instantes de simulacion no consumidos: 2918

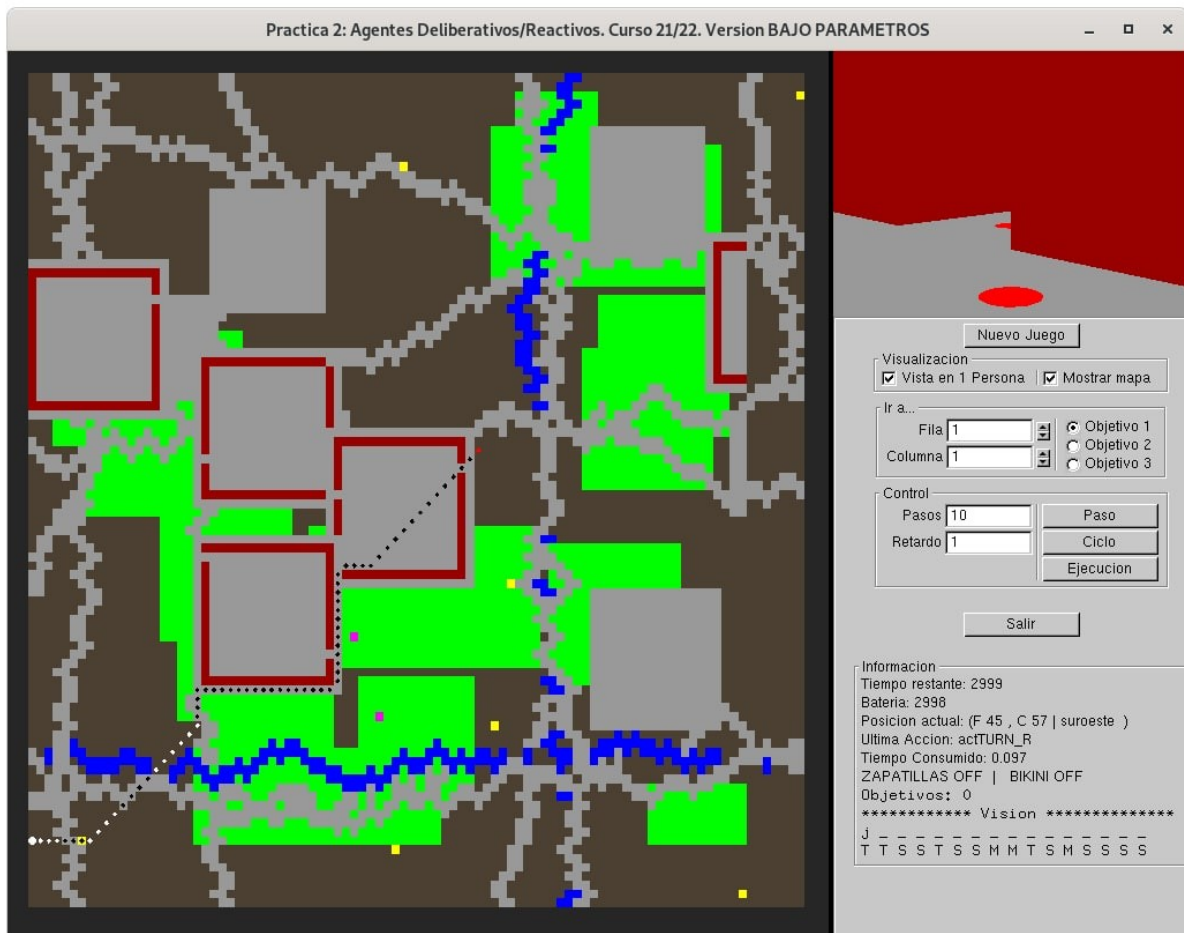
Tiempo Consumido: 1.65684

Nivel Final de Bateria: 2909

Plan: d d A A A A A A A A A A A A A A A A d A A A A I A A A A A A A A A A A
 A A d A d A A A A A A A A A A A A A A A i A i A A A d A A A A A A A d
 A A A A A A A i A A A A A A (longitud 81)

Ejemplo 7 (Heurística no admisible):

./practica2_MAL mapas/mapa100.map 1 2 45 57 3 89 3



Instantes de simulacion no consumidos: 2920

Tiempo Consumido: 0.100977

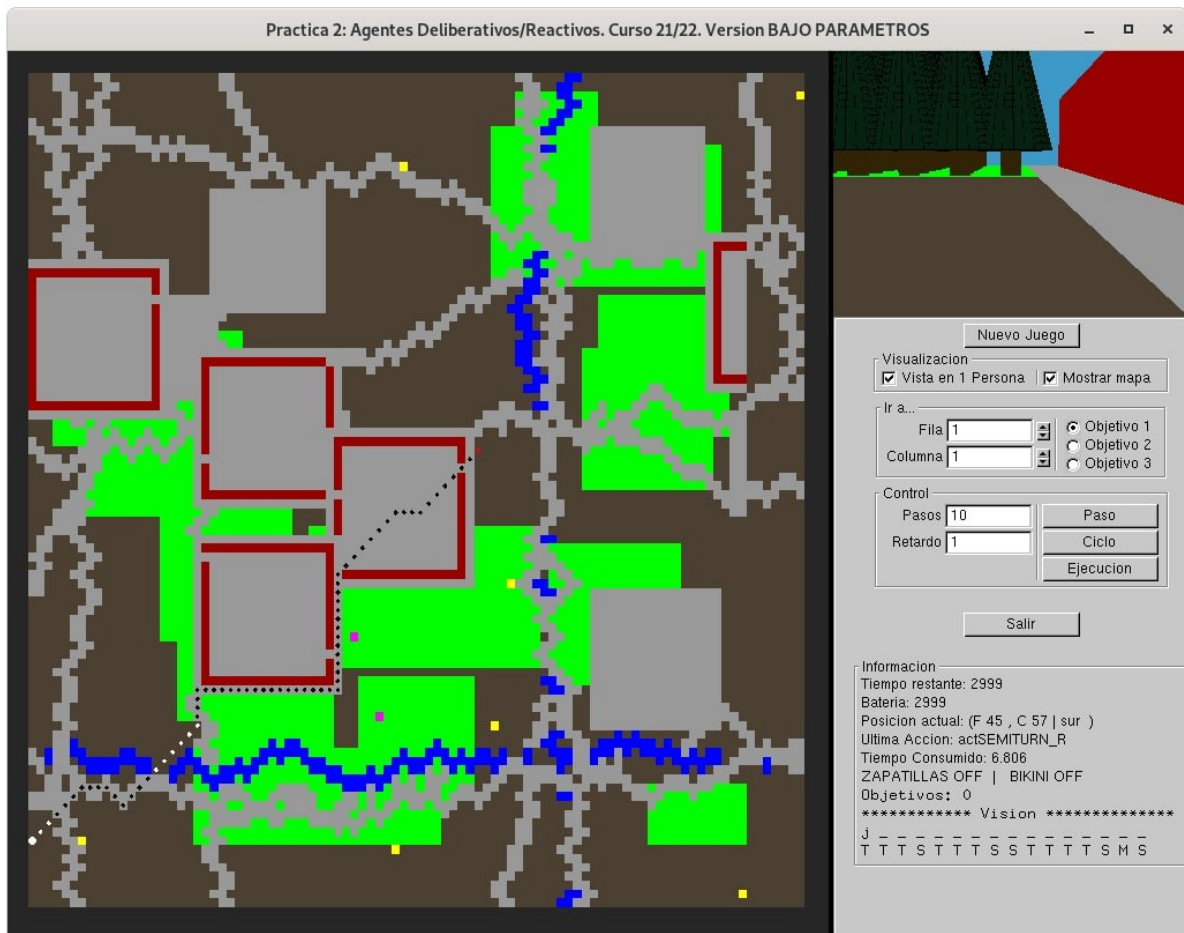
Nivel Final de Bateria: 2906

Plan: D A A A A A A A A A A A A A d A A A i A i A A A A A A A A A A A
A d A d A A A A A A A A A A A A A A A I A A A A d A A A A A A A A A
A A A A d A A A A A A A (longitud 79)

Destacar que el algoritmo de coste uniforme alcanza el objetivo con coste óptimo, aunque generalmente, será bastante más lento que el A*. En el ejemplo 8 vemos el camino encontrado al aplicar la búsqueda de costo uniforme en el mapa anterior. Vemos que el nivel final de batería vuelve a ser 2909, el óptimo encontrado por el A* con la heurística admisible. El plan puede variar mientras que el coste sea el mínimo, como ocurre en este caso.

Ejemplo 8 (Con coste uniforme):

./practica2 mapas/mapa100.map 1 2 45 57 3 89 3



Instantes de simulacion no consumidos: 2917

Tiempo Consumido: 6.80967

Nivel Final de Bateria: 2909

Plan: d d A A A A A A A d A A A i A A A A A A A i A A A A A A A A A A
A A A D A A A A A A A A A A A A A A A A I A A A A d A A A A A A A A
A D A A i A A A i A A A A A A (longitud 82)

Por último, queremos resaltar algunas de las propiedades que posee el algoritmo A* para que se tengan en cuenta a la hora de su uso a lo largo de esta práctica de forma muy resumida. Para ello haremos las siguientes consideraciones previas:

En un grafo dirigido con pesos, consideremos un nodo origen **O** y un nodo destino **G**. Llamamos **c** a la función de coste, definida para cualquier par de nodos conectados en el grafo. Sea **h** una función heurística en el grafo con respecto al nodo de destino **G**. Denotamos como **h*** a la función que, para cualquier nodo **N** del grafo proporciona el coste óptimo de llegar desde **N** hasta **G**.

1. Se dice que **h** es **admisble** si para cualquier nodo **N** del grafo, se tiene que $h(N) \leq h^*(N)$. Es decir, si **h** nunca sobreestima el coste de llegar al óptimo.

2. Se dice que **h** es **monótona** o **consistente** si $h(G) = 0$ y, para cualquier nodo **P** del grafo, y para cualquier nodo **H** hijo de **P**, se tiene que $h(P) - h(H) \leq c(P, H)$. Es decir, lo que decrece la heurística entre padre e hijo nunca sobrepasa el coste real de moverse del padre al hijo.

3. Si **h** es admisible, entonces el algoritmo A* , si encuentra un camino de **O** a **G**, el coste de dicho camino es óptimo.
4. Si **h** es monótona, cuando el algoritmo A* genera un nodo que ya estaba en cerrados, esa nueva ocurrencia del nodo no puede mejorar el coste de la primera ocurrencia.
5. Toda heurística monótona es admisible. En consecuencia, cuando una heurística es monótona, se puede aplicar el A* para encontrar el camino óptimo y no es necesario revisar los cerrados cuando salen repetidos.
6. Si **h** no es admisible, el algoritmo A* seguirá encontrando un camino de **O** a **G** si dicho camino existe, pero dicho camino no será necesariamente óptimo en coste.
7. Si **h** es admisible pero no monótona, el algoritmo A* puede generar nodos que ya estaban cerrados pero que tengan un mejor coste que el cerrado actual. En consecuencia, para garantizar que se encuentra el camino óptimo es necesario reemplazar dicho cerrado y, recursivamente, todos aquellos nodos que hayan sido expandidos a partir de dicho cerrado.
8. La distancia de Manhattan o la euclídea en un mundo cuadrado con las 8 orientaciones que se proponen en la práctica no serían monótonas ni admisibles.

4. Nivel 3

El objetivo de este nivel es parecido al objetivo que se planteó en la práctica 1, definir un comportamiento para un agente que permita descubrir el contenido del mapa en el que se encuentra partiendo de una situación donde lo desconoce por completo. A diferencia de la práctica anterior, en este comportamiento se permiten definir comportamientos deliberativos, de hecho, la intención es que sea este comportamiento la base de la estrategia de descubrimiento.

La forma en la que se evaluará es semejante **PERO NO IGUAL** a como se hacía en un mapa concreto en una configuración concreta en la práctica anterior. Las principales diferencias con respecto a la evaluación anterior son las siguientes:

1. Los fallos en el reconocimiento del mapa penalizarán. Se considera un fallo un valor consignado en una casilla de **MapaResultado**, distinto de '?' que no coincida con el valor de dicha casilla en el mapa original.
2. Por debajo de un 20% de reconocimiento del mapa, el valoración asociada es 0 puntos.
3. Por encima del 90% de reconocimiento, la valoración asociada es de 2 puntos.
4. Entre el 20 y 90 por ciento se realiza una interpolación líneal para determinar la calificación.

5. En la evaluación se aplicará sobre varias ejecuciones distintas (pudiendo ser sobre mapas distintos) y la valoración final para este nivel de una práctica concreta será la media sobre esas ejecuciones.

6. Las ejecuciones que no consigan terminar de forma normal la simulación¹ tendrán penalización y entrarán a formar parte en el cálculo de la media con valor **-0,1**.

Así, de forma concreta, en la siguiente tabla se muestran algunas calificaciones en función del porcentaje del mapa descubierto.

Porcentaje	<=20%	30%	55%	75%	>=90%
Puntuación	0	0,29	1,00	1,57	2

5. Nivel 4

Como ya se menciona en el guion de la práctica, en el nivel 4 se espera que el estudiante defina un agente que combine comportamientos reactivos y deliberativos con la intención de conseguir el máximo número de objetivos bajo las condiciones definidas en cuanto a número de instantes de simulación, valor de la batería y tiempo. A continuación vamos a mostrar algunos problemas usando algunos de los mapas proporcionados para ilustrar el como se valorarán los resultados obtenidos por el estudiante. La lista es la siguiente:

Id	Definición del Problema
P1	./practica2SG mapas/ mapa30.map 1 4 4 4 6 3 3 5 10 16 14 18 10 21 4 7 4 24 4 12 3 6 15 25 18 8 19 15 14 23 15 26 8 10 24 19 26 25 7 16 11 22 15 20 15 22 19 10 20 4 13 26 24 9 6 26 10 17 19 25 13 24 20 26 19 12 18 8 23 9 13 6 5 8 16 12 5 3 14 11 22 11 8 6 17 7 4 21 3 23 4 15 5 7 23 21 19 4 15 6 13 24 17 6 26 5 4 24 10 16 17 13 20 22 9 26 22 22 24 14 4 24 26 7 18 6 21 9 9 18 6 19 15 16 21 3 14 13 10 25 13 17 24 7 20 14 14 9 21 5 18 20 20 19 10 5 18 18 21 9 22 20 19 7 15 26 20 10 17 17 19 8 23 8 9 9 5 20 3 20 11 6 23 16 8 26 14 17 4 8 25 14 13 25 14 8 5 20 21 4 18 14 25 21 5
P2	./practica2SG mapas/ mapa50.map 1 4 28 25 3 39 45 9 16 38 13 27 23 31 18 45 31 35 7 12 6 40 7 20 6 10 25 41 30 14 31 26 24 38 26 38 20 44 14 17 40 45 3 4 9 33 44 17 3 3 11 42 13 26 18 38 25 33 26 46 46 36 14 36 31 17 34 8 22 44 41 16 11 44 17 29 32 42 21 46 19 40 34 45 24 46 7 44 32 21 30 14 39 15 22 11 9 13 27 20 8 45 5 6
P3	./practica2SG mapas/ mapa75.map 1 4 11 47 0 59 68 3 32 14 25 57 20 41 35 47 27 38 71 68 47 53 10 18 10 60 11 16 38 62 45 22 47 39 37 57 39 57 30 66 21 26 60 68 5 6 14 49 66 26 4 4 17 63 19 40 28 57 38 50 39 69 69 54 21 55 47 26 51 12 33 66 62 24 17 67 26 51 71 44 49 64 32 69 29
P4	./practica2SG mapas/ mapa100.map 1 4 63 31 3 3 3 79 91 19 33 76 27 55 47 62 36 51 95 91 63 71 14 24 13 80 15 21 51 83 61 29 63 52 49 77 52 76 40 89 28 35 80 91 6 94 52 8

1 Se considerarán finalizaciones anormales aquellas que hagan que la ejecución termine antes que alguna de las condiciones que se especifican en la práctica (consumir el número de instantes de simulación, consumir los puntos de batería o consumir los 300 segundos de pensar) se satisfaga. Morir por caer por un precipicio también tiene la consideración de finalización anormal.

	19 66 89 34 6 6 23 85 26 53 37 76 51 66 53 3 43
P5	./practica2SG mapas/ pinkworld.map 1 4 68 14 1 3 3 41 35 47 27 38 71 68 47 18 10 60 11 30 9 66 21 68 5 71 39 6 14 49 66 4 17 40 28 50 39 69 69 12 33 66 62 67 26 51 71 44 49 64 32 69 29 68 36
P6	./practica2SG mapas/ islas.map 1 4 47 53 2 74 47 46 42 71 56 83 52 58 65 85 43 92 39 81 68 91 48 21 95 92 14 88 64 43 61 28 78 30 44 22 18 27 55 41 16 90 10 12 49 76 68 38 74
P7	./practica2SG mapas/ marymonte.map 1 4 66 38 0 59 64 41 35 12 65 71 68 47 53 10 45 69 8 6 37 68 59 36 48 11 13 70 70 8 67 59 10 34 70 33 4 66 70 37 22 46 71 70 18
P8	./practica2SG mapas/ medieval.map 1 4 3 3 2 9 20 13 48 9 87 84 39 91 19 33 76 87 39 57 29 84 53 96 96 63 3 88 54 32 44 19 75 29 95 96 48 21 33 87 41 9 20 13 48 9 87 84 39 91 19 33 76 87 39

Como se puede observar, los 8 ejemplos se ejecutan usando la versión del software que no usa gráficos (*practica2SG*).

También vamos a definir 3 niveles (**Bajo**, **Normal** y **Alto**) para que el estudiante pueda estimar como es el funcionamiento de su agente en base a los objetivos alcanzados. Evidentemente, sabemos que el resultado exacto no depende solo del mapa, sino de la configuración inicial completa, pero si que servirá como guía para que el estudiante pueda situar sus resultados con los esperables en cada categoría. En la siguiente tabla se muestran los intervalos asociados a cada problema.

	Bajo	Normal	Alto
P1	<30	[30,90)	>=90
P2	<18	[18,50)	>=50
P3	<8	[8,40)	>=40
P4	<10	[10,25)	>=25
P5	<8	[8,20)	>=20
P6	<3	[3,10)	>=10
P7	<4	[4,20)	>=20
P8	<2	[3,12)	>=12

En el cálculo numérico de la calificación para el nivel 4 se procederá de la siguiente forma:

1. Se obtendrán **3 puntos** en las ejecuciones que consigan el nivel **Alto**.
2. Aquellas ejecuciones que consigan el nivel **Normal**, tendrá una calificación entre **1 y 3 puntos**. Su valoración concreta será el resultante de la interpolación lineal de ese valor en función de intervalo inferior y superior que defina la etiqueta **Normal**.
3. De igual manera, para los que estén bajo la etiqueta **Bajo** tendrán una calificación entre **0 y 1 punto** que será la interpolación lineal entre 0 y el valor considerado como límite superior de esa etiqueta.

4. En la evaluación se aplicarán varias ejecuciones distintas (pudiendo ser sobre mapas distintos) y la valoración final para este nivel de una práctica concreta será la media sobre esas ejecuciones.
5. Las ejecuciones que no consigan terminar de forma normal la simulación² tendrán penalización y entrarán a formar parte en el cálculo de la media con valor **-0,1**.

2 Se considerarán finalizaciones anormales aquellas que hagan que la ejecución termine antes que alguna de las condiciones que se especifican en la práctica (consumir el número de instantes de simulación, consumir los puntos de batería o consumir los 300 segundos de pensar) se satisfaga. Morir por caer por un precipicio también tiene la consideración de finalización anormal.