

## INTELIGENCIA ARTIFICIAL

CURSO 2020-21

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

APELLIDOS Y NOMBRE	Lugli, Valentino		
GRUPO TEORÍA	C	GRUPO PRÁCTICAS	C1

#### Instrucciones iniciales

En este formulario se proponen preguntas que tienen que ver con ejecuciones concretas del software desarrollado por los estudiantes. También aparecen preguntas que requieren breves explicaciones relativas a como el estudiante ha hecho algunas partes de esa implementación y que cosas han tenido en cuenta.

En las preguntas relativas al funcionamiento del software del alumno, estas se expresan haciendo uso de la versión de invocación en línea de comandos cuya sintaxis se puede consultar en el guion de la práctica.

El estudiante debe poner en los recuadros la información que se solicita.

En los casos que se solicita una captura de pantalla (**ScreenShot**), extraer la imagen de la ejecución concreta pedida donde aparezca la línea de puntos que marca el camino (justo en el instante en el que se construye obtiene el plan). Además, en dicha captura debe aparecer al menos el nombre del alumno. Ejemplos de imágenes se pueden encontrar en [Imagen1](#) y en [Imagen2](#).

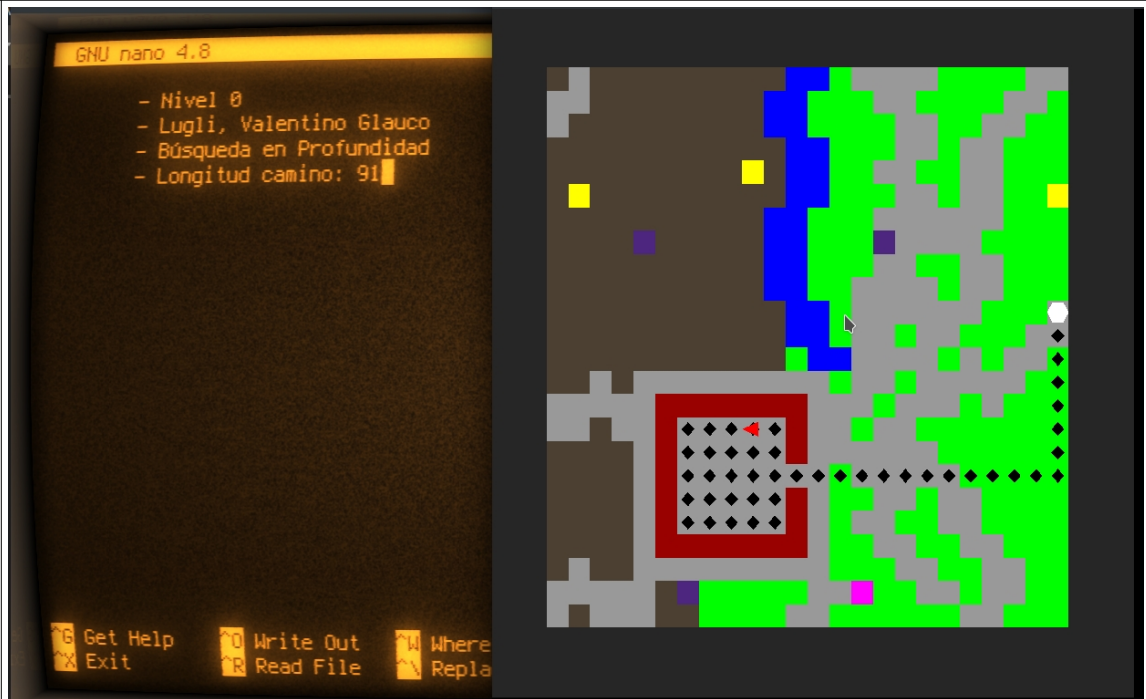
**Enumera los niveles presentados en su práctica (Nivel 0, Nivel 1, Nivel 2, Nivel 3, Nivel 4):**

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

## Nivel 0-Demo

(a) Rellena los datos de la tabla con el resultado de aplicar

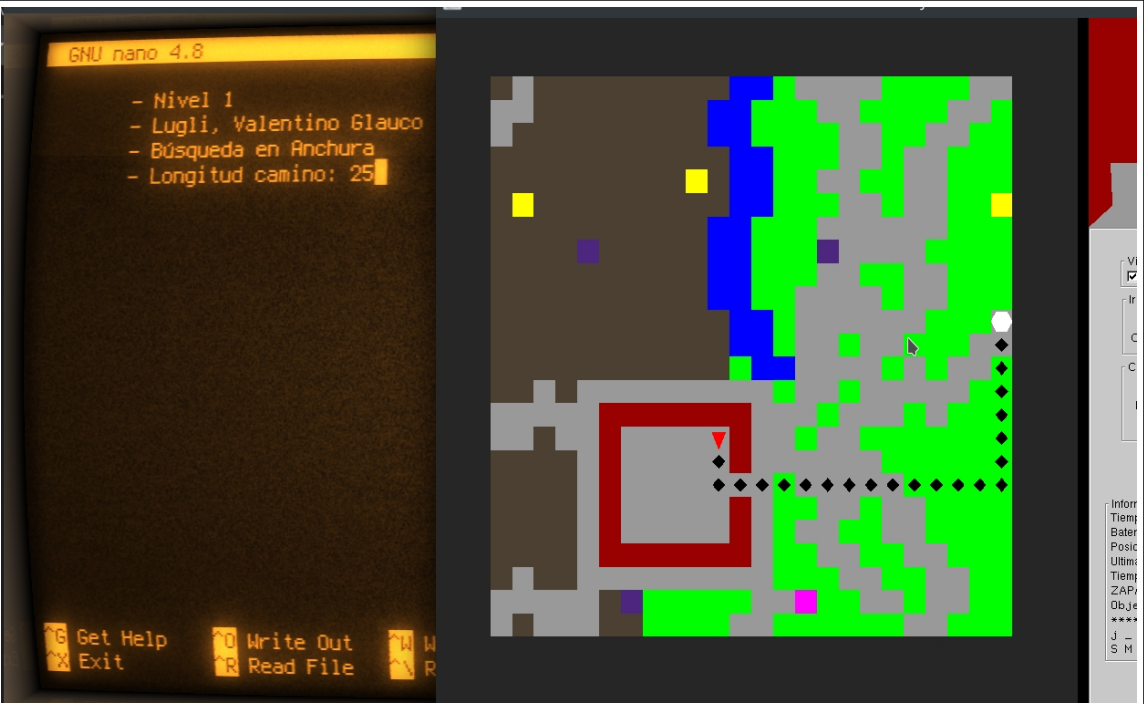
**Belkan mapas/mapa30.map 1 0 18 13 3 13 26**



	<b>Instantes de simulación no consumidos</b>																												2908										
<b>Tiempo Consumido</b>																																		0.00489					
<b>Nivel Final de Batería</b>																																		1818					
<b>Plan</b>	A A A A I A A A A I A A A A I A A A A D D A A A A D A A A A																																						
	D A A A A D A A A D A A A I I A A I A A I A																																						
	A I A I A A D D A A D A D A D A A A A A A A A A A A A A A A A A																																						
	A I A A A A A A A																																						

**Nivel 1-Óptimo en número de pasos**

- (a) Rellena los datos de la tabla con el resultado de aplicar  
**Belkan mapas/mapa30.map 1 1 18 13 3 13 26**

ScreenShot		
	Instantes de simulación no consumidos	2974
	Tiempo Consumido	0.06136
	Nivel Final de Batería	1884
	Plan	I A A I A A A A A A A A A A A I A A A A A A A

(b) Rellena los datos de la tabla con el resultado de aplicar

**Belkan mapas/mapa30.map 1 1 20 11 3 21 6**

ScreenShot		
	Instantes de simulación no consumidos	2974
	Tiempo Consumido	0.111794
	Nivel Final de Batería	2971
	Plan	D D A A A A D A A A A D A A A A A A A A A D A A A

## Nivel 2-Óptimo en coste 1 objetivo

(a) Indica el algoritmo implementado para realizar este nivel

Algoritmo A\*

(b) Si usaste A\*, indica la heurística utilizada

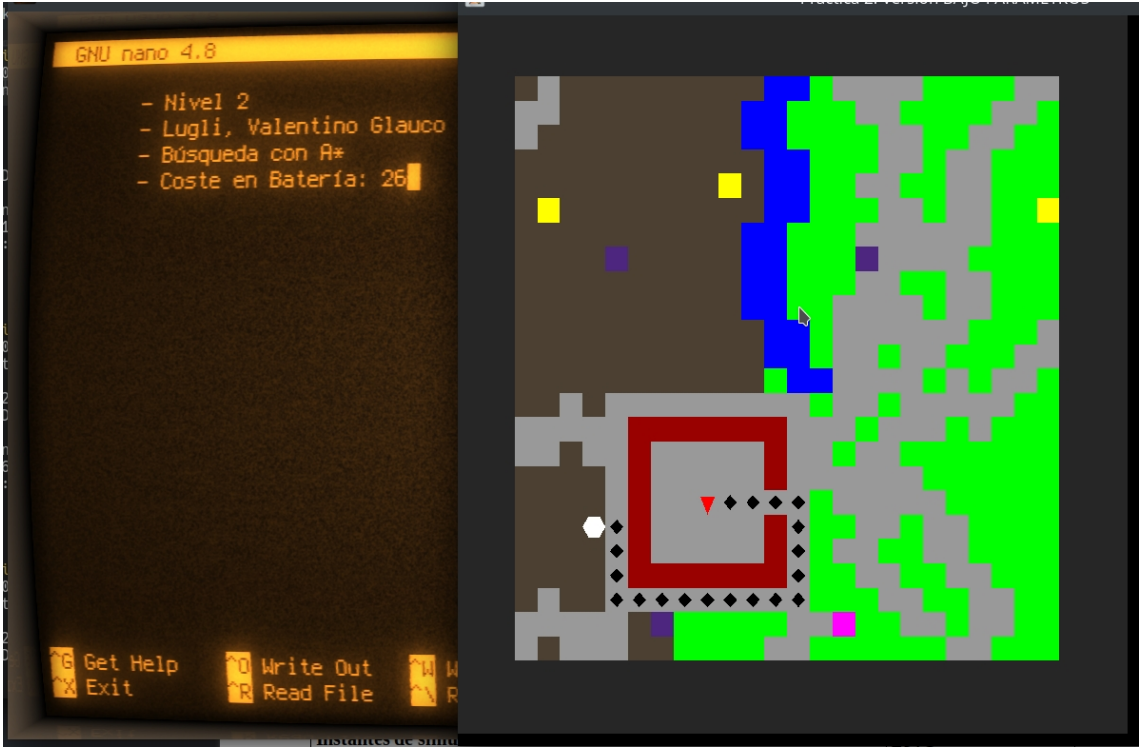
Distancia Manhattan

(c) ¿cambió el concepto de estado con respecto al usado en el nivel 0 y 1? Si la respuesta es afirmativa, explica brevemente en que consistió el cambio.

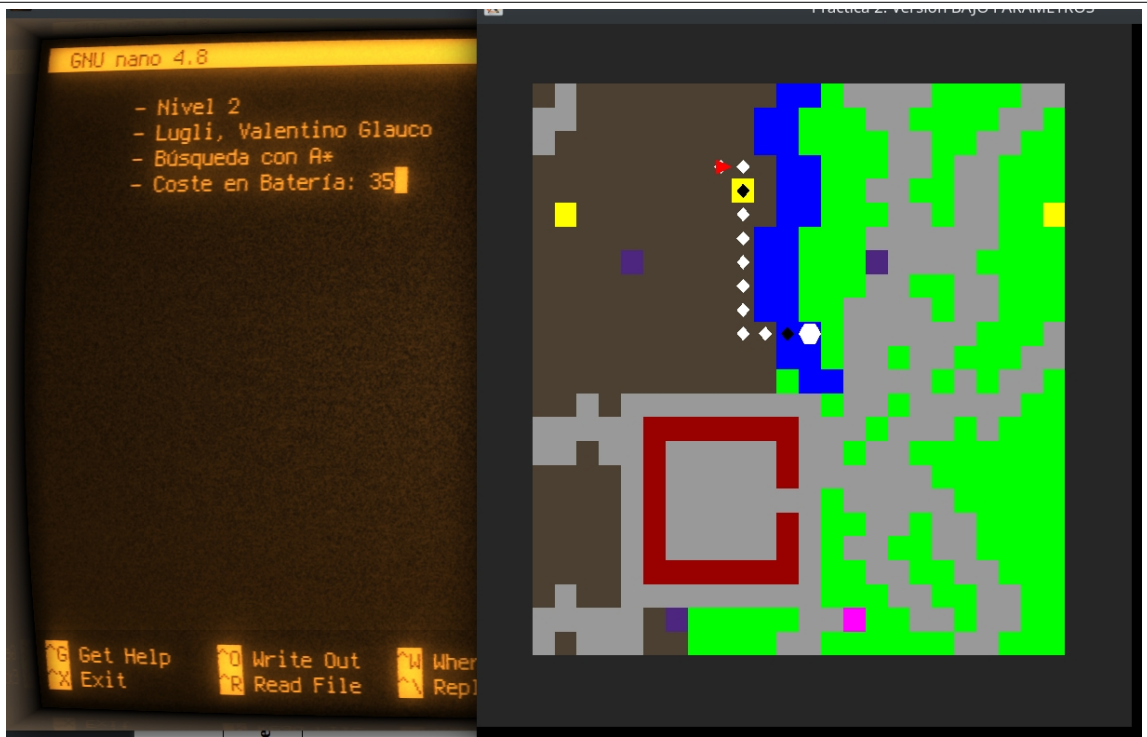
Si, se creó una clase nueva aStarNode, donde el estado de cada nodo ahora almacena además de la posición y orientación del agente:

- El coste total,  $f(n)$ , que es una suma de lo que se ha recorrido, la longitud en acciones,  $g(n)$  y el coste esperado calculado por la heurística para llegar al objetivo,  $h(n)$ .
- Si en ese nodo se posee Bikini o Zapatillas
- Se añadió lo necesario para que la comparación de nodos de la lista de Cerrados se ajustara al cambio.

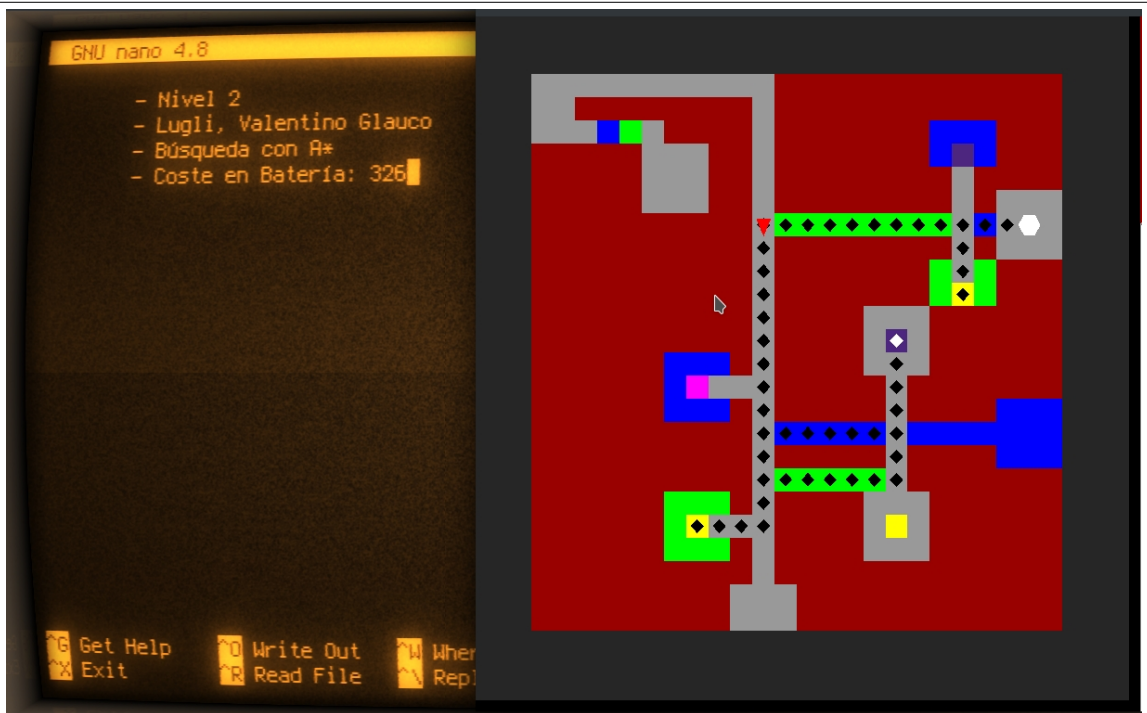
(d) Rellena los datos de la tabla con el resultado de aplicar  
**Belkan mapas/mapa30.map 1 2 20 11 3 21 6**

ScreenShot			
Instantes de simulación no consumidos		2973	
Tiempo Consumido		0.006035	
Nivel Final de Batería		2974	
Plan	I I A A A A D A A A A D A A A A A A A A D A A A I A		

(e) Rellena los datos de la tabla con el resultado de aplicar  
**Belkan mapas/mapa30.map 1 2 6 10 1 13 15**

ScreenShot		
	Instantes de simulación no consumidos	
	Tiempo Consumido	
	Nivel Final de Bateria	
	Plan	A A D A A A A A A I A A A

(f) Rellena los datos de la tabla con el resultado de aplicar  
**Belkan mapas/scape.map 1 2 9 13 1 9 25**

ScreenShot		
	Instantes de simulación no consumidos	
	Tiempo Consumido	
	Nivel Final de Batería	
	Plan	D A A A A A A A A A A A A A D A A A I I A A A I A A A A D A A A A A A I A A A A I I A A A A A A D A A A A A A D A A A A A A A A A A D A A A A A A A A A D A A A I I A A A D A A A



### Nivel 3-Óptimo en coste 3 objetivos

(a) Indica el algoritmo implementado para realizar este nivel

Algoritmo A\*

(b) Si usaste A\*, describe la heurística utilizada. ¿Es admisible?

Si, se utilizó la misma distancia Manhattan pero multidesestino, es decir, es la distancia Manhattan de la posición del nodo hasta el primer objetivo que tenga ese nodo, luego de ese objetivo al segundo y del segundo al tercero, si es que posee en ese nodo segundo o tercer objetivo, se calcula para la cantidad de objetivos que tenga faltantes.

Es admisible porque la heurística no sobreestima el camino, es en realidad, el coste mínimo que se podría obtener asumiendo que no hubiera obstáculos entre el nodo y sus objetivos y todo el terreno tuviera un coste en batería unitario.

(c) ¿has usado un concepto de estado diferente al utilizado en los algoritmos empleados anteriormente? Si la respuesta es "Sí", describe brevemente cómo cambió y en qué afecta al algoritmo usado.

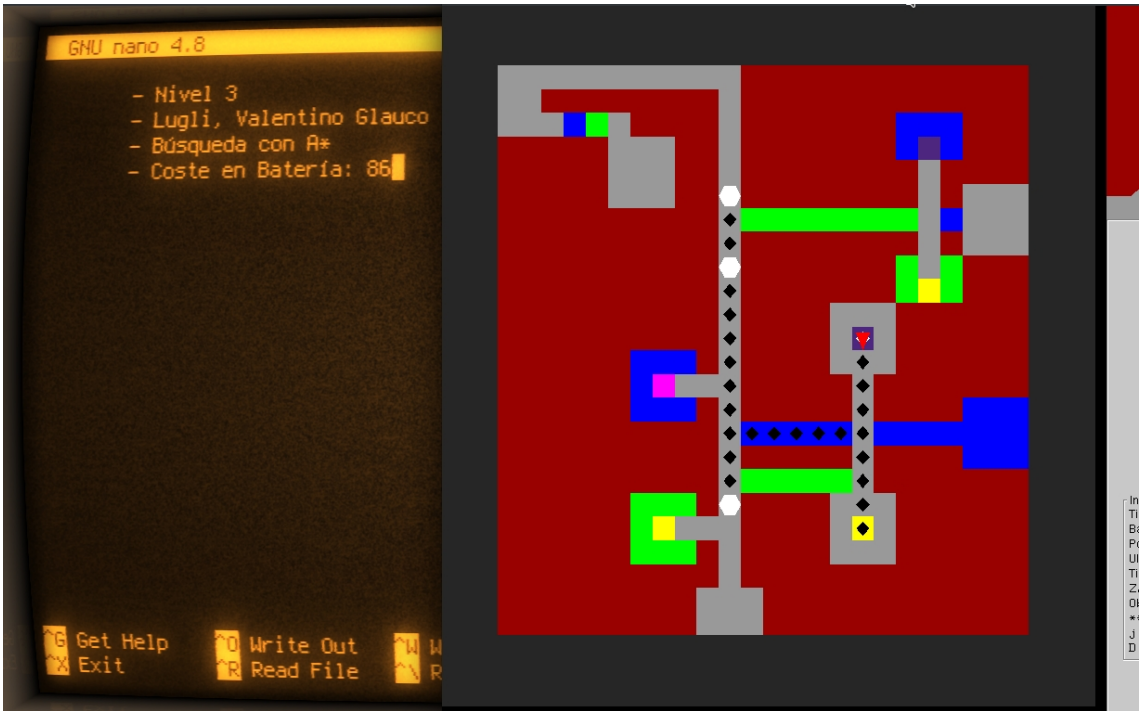
Si. La clase aStarNode ahora también mantiene además de lo mencionado en el nivel 2, una lista de los objetivos que el nodo posee, tomándose como que el objetivo al frente de esta lista es el objetivo al que el nodo está dirigiéndose actualmente.

Se modificaron las funciones de comparación para tomar en cuenta ahora también la cantidad de objetivos que posee cada nodo, y en caso de tener la misma cantidad, en qué orden están los objetivos en la lista.

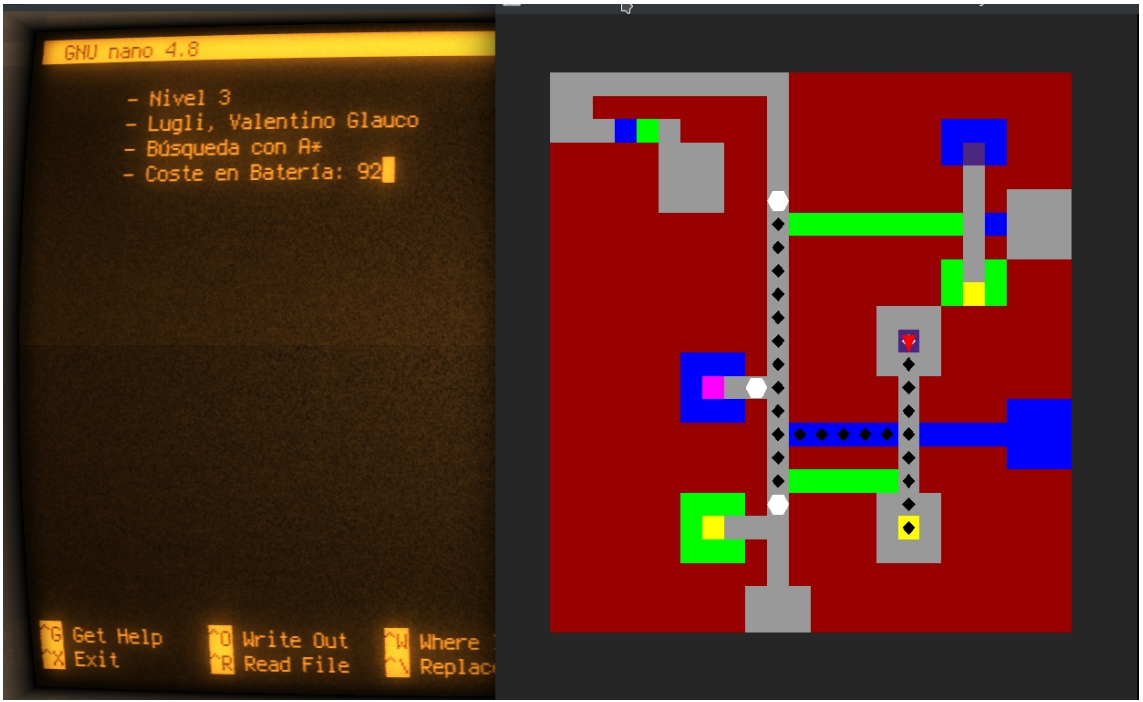
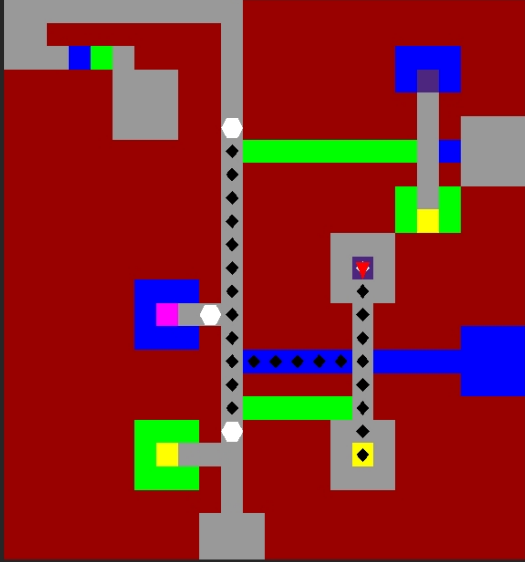
(d) Incluye aquí todos los comentarios sobre las dificultades que te llevó la implementación de este nivel.

- Pensar en cómo adaptar de un solo objetivo a tres objetivos, qué añadir al aStarNode y cómo adaptar la heurística.
- Dificultades para la comparación de nodos de cerrados, se tuvo que crear una función auxiliar para ello porque el if estaba creciendo a un tamaño exorbitante.
- Añadir lo necesario para tomar en cuenta el orden de los objetivos, es decir, generar más hijos en un nodo para que cada trío de hijos fuese a un objetivo diferente, así podrían obtener el coste más bajo de batería y no necesariamente es tomando que el primer objetivo sea el más cercano.

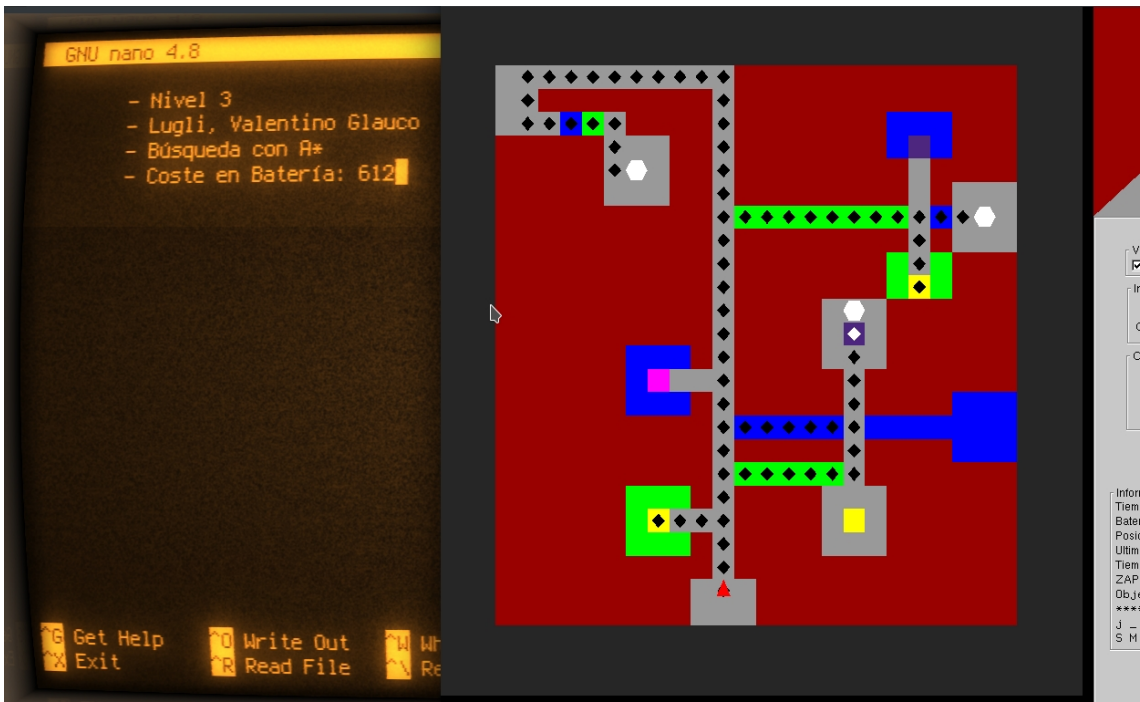
(e) Rellena los datos de la tabla con el resultado de aplicar  
**Belkan mapas/scape.map 1 3 13 19 2 8 13 21 13 11 13**

ScreenShot		
	Instantes de simulación no consumidos	2958
	Tiempo Consumido	0.887694
	Nivel Final de batería	2914
	Plan	A A A A A A A A A I I A A A A I A A A A A A I A A A I I A A A A A A A A A A A A A

(f) Rellena los datos de la tabla con el resultado de aplicar  
**Belkan mapas/scape.map 1 3 13 19 2 8 13 21 13 16 12**

ScreenShot		
		
Instantes de simulación no consumidos		2952
Tiempo Consumido		0.932403
Nivel Final de Batería		2908
Plan	A A A A A A A A A A I I A A A A I A A A A A A I A A A I I A A A A A I A I I A I A A A A A A A A	

(g) Rellena los datos de la tabla con el resultado de aplicar  
**Belkan mapas/scape.map 1 3 26 13 0 9 25 7 9 13 19**

ScreenShot			
	Instantes de simulación no consumidos		2839
	Tiempo Consumido		53.129
	Nivel Final de Batería		2388
	Plan	A A A A I A A A I I A A A I A A A A A A A A A A A A A A A A A A I A A A A A A A A A I A A I A A A A D A A I A I I A D A A I A A A A D A A D 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 I A A A A A A I I A A A A A A A D A A A A A A D A A A A A A A A A A A A A D A A A I I A A A D A A A	

## Nivel 4-Reto

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

Algoritmo A\*

- (i) ¿Has incluido dentro del algoritmo de búsqueda usado en el nivel 4 que si pasas por una casilla que da las zapatillas o el bikini, considere en todos los estados descendientes de él que tiene las zapatillas y/o el bikini? En caso afirmativo, explicar brevemente cómo.

Si. El Agente tiene dos miembros booleanos que en el método think comprueban por cada “tick” que si la casilla donde están es una casilla de Bikini o de Zapatillas, cuando se llama al Algoritmo A\*, al constructor del nodo raíz se le pasa el estado de esos dos miembros de la clase para que de una vez tome en cuenta si está equipando el Bikini o las Zapatillas, por lo tanto una vez comienza el bucle principal del A\*, ya se está tomando en cuenta eso.

- (j) Explica brevemente qué política has seguido para abordar el problema de tener 3 objetivos.

Se creó un nuevo miembro de la clase que es una lista que mantiene los objetivos actuales, es diferente de la lista que se encuentra originalmente implementada; ya que la lista original obtiene los 3 objetivos y siempre se está sobrescribiendo en cada llamada a think. Esta segunda lista copia los objetivos de la primera y se va reduciendo hasta que esté vacía, lo que implica que el agente ya ha pasado por estos tres objetivos. En este momento vuelve a copiar los objetivos de la primera para planificar la ruta nuevamente.

Los objetivos ahora son de un struct objective que mantiene a parte del estado, mantienen también la distancia del Agente a ellos, de esta manera la lista de objetivos actuales se ordena con el objetivo más cercano de primero; además se toma en cuenta que es posible que el Agente cuando vaya hacia una estación de carga en su ruta pase por un objetivo, y lo quita de los objetivos actuales.

- (k) ¿Bajo qué condiciones replanifica tu agente?

El agente replanifica cuando:

- Al descubrir partes nuevas del mapa, en sus sensores de cercanía detecta que hay casillas con Muros, Precipicios, Agua o Bosque, entonces renta replanificar para encontrar una mejor ruta.
- Cuando llega a un objetivo, replanifica para el siguiente objetivo en la lista de los más cercanos a él.
- Cuando el nivel de batería es menor de 1500 unidades, replanifica para ir a una estación de recarga si ha encontrado alguna y también cuando ya se ha recargado, replanifica para

retomar el objetivo por dónde se quedó.

- Cuando un aldeano se encuentra en su camino y hay una colisión, replanifica para pasarle por un lado.

(l) ¿Qué coste le has asignado a la casilla desconocida en la construcción de planes cuando el mapa contiene casillas aún sin conocer?. Justifica ese valor.

La casilla desconocida tiene un coste de 1, se prefirió que el Agente sea curioso y si se encuentra con un terreno costoso para llegar a un objetivo, es preferible asumir que puede que haya una mejor ruta por el terreno que no es conocido de momento, por lo tanto, el Agente explorará más hacia lo desconocido. Como nota, como se sabe que los mapas siempre tendrán 3 casillas de precipicios en los bordes, esto es rellenado automáticamente en el constructor del Agente, esto evita que el agente dé muchas vueltas intentando buscar una mejor ruta por los bordes, pero se mantenga curioso internamente en el mapa.

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

Si, en cada llamada a think se actualiza un miembro de la clase que mantiene el nivel de batería, una vez que este nivel es menor que 1500 unidades, el Agente entonces escanea el mapa conocido, guardando en una lista las estaciones de recarga con el frente de la lista la más cercana al mismo, luego, se replanifica para que ahora el objetivo sea ir hacia esa estación de recarga; una vez recargando, cuando se llega a 2000 unidades de batería, el Agente retoma el objetivo por donde se había quedado.

(n) Añade aquí todas los comentarios que desees sobre el trabajo que has desarrollado sobre este nivel, qué consideras con 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.

- Para esquivar los Aldeanos, se hace que si hay una colisión con ellos, en el mapa se calcula la posición de los mismos y se marca temporalmente como si fuese un muro, de esta manera el A\* replanifica pasándole por un lado, luego esas secciones del mapa se restauran a como estaban antes por lo tanto no hay una modificación visual del mapa porque todo sucede dentro del think. Para esto se crearon distintos métodos y los aldeanos se guardan en una lista de tipo enemy, el cual dentro mantiene el estado donde están y también en que sitio del sensor de superficie fueron vistos, así es como se marcan en el mapa y luego utilizando el sensor de terreno se restaura el mapa.
- Se crearon nuevos structs, como el anterior mencionado de los aldeanos, también están el de tipo objective para mantener dentro los objetivos como tal y también dónde se encuentran las estaciones de recarga.

(o) Rellena los datos de la tabla con el resultado de aplicar

**Belkan 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**

<b>Instantes de simulación no consumidos</b>	0
<b>Tiempo Consumido</b>	153.357
<b>Nivel Final de Batería</b>	1394
<b>Objetivos</b>	18

(p) Rellena los datos de la tabla con el resultado de aplicar

**Belkan 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**

<b>Instantes de simulación no consumidos</b>	0
<b>Tiempo Consumido</b>	74.1579
<b>Nivel Final de Batería</b>	1261
<b>Objetivos</b>	14