El laberinto

Un laberinto se conforma por un conjunto de caminos que intentan confundir a quienes se adentran en él. Tenemos dos variantes principales que son: el laberinto clásico que es aquel que te obliga a recorrer todas sus vías para llegar a la salida (no ofrece la posibilidad de tomar caminos alternativos). Y el laberinto "mazes" que ofrece caminos alternativos al recorrer su interior, siendo uno o varios los que lleven a la salida del mismo, mientras que el resto de los caminos no. Los laberintos pueden tener principalmente forma circular, cuadrada o rectangular.

El reto consiste en escribir un programa de cómputo que recorre un *laberinto rectangular* y encuentre la salida en el menor número posible de movimientos.

Maze-challenge es un programa de cómputo que ofrece una interfaz para interactuar con él y será proporcionado al participante del reto. El ***Jugador*** es el programa de cómputo implementado por el participante del reto que intenta encontrar la salida del laberinto.

A continuación se describe con detalle la interfaz proporcionada por el programa de cómputo Maze-challenge.

El nombre de la interfaz es **Maze,** se recibe con el nombre de game y contiene las siguientes funciones:

start()

Este servicio deberá ejecutarse para indicarle al ***Laberinto*** que el ***Jugador*** ha iniciado su recorrido por el mismo. A partir del momento que invoque este servicio, iniciará el conteo de los movimientos realizados que le tome al ***Jugador*** encontrar la salida. **Nota**: esta función se ejecuta automáticamente al dar click en el botón "Play" en caso que intente ejecutar nuevamente la función, es posible que reciba un error/excepción o simplemente el programa ***Maze-challenge*** no funcione adecuadamente.

watchLocation() - game.watchLocation()

Este servicio permite indicarle al ***Jugador*** las opciones de movimiento de acuerdo a su posición actual. El servicio retorna como respuesta un número entero que representa las opciones de movimiento que el ***Jugador*** tiene desde su posición actual.

Este servicio retorna como respuesta un número entero que representa las opciones de movimiento que el ***Jugador*** tiene desde su posición actual de acuerdo a la Tabla 1.

TABLA 1. RELACIÓN DE VALORES Y DIRECCIÓN DE MOVIMIENTO PERMITIDAS.

Valor	Norte	Sur	Este	Oeste
0	Si	Si	Si	Si
1	Si	No	Si	Si
2	Si	Si	No	Si
3	Si	No	No	Si
4	No	Si	Si	Si
5	No	No	Si	Si
6	No	Si	No	Si

7	No	No	No	Si
8	Si	Si	Si	No
9	Si	No	Si	No
10	Si	Si	No	No
11	Si	No	No	No
12	No	Si	Si	No
13	No	No	Si	No
14	No	Si	No	No

En la columna de Valor, debe buscar el valor recibido como respuesta, por ejemplo el 6. Si se revisan el resto de las columnas (Norte, Sur, Este y Oeste), va encontrar (Si) y (no). El Si indica que el movimiento hacia esa dirección está permitido, mientras que el No indica que el movimiento hacia esa dirección NO está permitido. En el ejemplo para el valor 6, está permitido moverse únicamente hacia Sur u Oeste.

En la Figura 1 se muestra gráficamente lo anteriormente descrito, pero ahora para el ejemplo de recibir como respuesta el 13, suponiendo que el ***Jugador*** está ubicado en la **X**, el Norte siempre estará hacia arriba, el Sur hacia abajo, el Este hacia la derecha y el Oeste a la izquierda. Entonces recibiendo como respuesta del servicio el número 13 significa que solo puede moverse hacia la dirección Este.

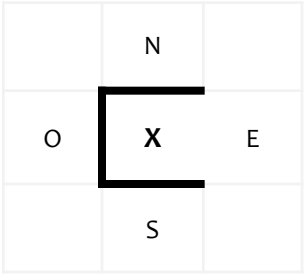

Figura 1. Demostración gráfica de las opciones de movimiento para un valor de 13.

async move(MovementDirection) - await game.move(MovementDirection.NORTH)

Este servicio permite al ***Jugador*** cambiar su posición dentro del laberinto. El servicio recibe como parámetro la dirección (utilizar el enumerador MovementDirection) hacia donde se desea mover, teniendo como opciones NORTH, SOUTH, EAST y WEST. El servicio retorna como respuesta un número entero de acuerdo indicado en la Tabla 2.

Tabla 2. Relación de valor retornado por el servicio mover y su significado.

Valor	Descripción
0 al 14	Indicará que el movimiento se realizó satisfactoriamente, y el valor recibido corresponde a las opciones de movimiento que el ***Jugador*** tiene desde la nueva ubicación (ver Tabla 1).

-1	Indicará que el movimiento solicitado no está permitido desde la posición actual del ***Jugador***. Tendrá que validar con el servicio Consultar si la opción enviada es válida.
15	Indicará que el movimiento realizado lo llevó al final del laberinto, lo cual significa que ha encontrado la salida y ya no es necesario realizar más movimientos.

isDone() - game.isDone()

Este servicio permite consultar si el Jugador se encuentra en el final del recorrido del laberinto, es decir, si está en la salida. El servicio retorna como respuesta un booleano que en caso de ser verdadero (true) es porque el Jugador está en la salida, en caso contrario, es necesario seguir el recorrido hasta encontrar la salida.

Consideraciones

El laberinto tendrá una forma rectangular con las dimensiones MxN en donde M representa el ancho y N el alto del laberinto. Al ejecutar la funcion start() de manera automática el ***Laberinto*** ubicará al ***Jugador*** en las coordenadas de inicio Xi, Yi. La información sobre las dimensiones MxN y las coordenadas de inicio Xi, Yi NO están disponibles para el ***Jugador***, por lo cual no debe suponer que debe moverse en una dirección en particular.

El enumerador MovementDirection tiene los siguientes valores:

SOUTH = 1

EAST = 2

NORTH = 4

WEST = 8