-Memoria Practica 1-

*Algoritmo A**

En esta primera práctica se implementa el algoritmo A* en JavaScript para encontrar el camino óptimo entre un punto de inicio y un punto de destino en una cuadrícula de MxN.

En cuanto al lenguaje utilizado, el código está escrito en JavaScript, un lenguaje de programación ampliamente utilizado en el desarrollo web. JavaScript se eligió por su capacidad para interactuar con el DOM (Document Object Model) y permitir la manipulación dinámica de elementos HTML, lo que facilita la visualización de la cuadrícula y la interacción del usuario con ella.

La implementación del algoritmo A* comienza con la definición de nodos en la cuadrícula, representados por la clase `Node`. Cada nodo tiene coordenadas `x` e `y`, así como propiedades para almacenar el costo acumulado desde el nodo inicial (`g`), el costo estimado hasta el nodo final (`h`), y el costo total (`f`). Además, hay propiedades booleanas para indicar si el nodo está prohibido, es un punto de inicio o destino, o si contiene peligros o montañas.

La función principal del algoritmo A* es `search`, que realiza la búsqueda del camino óptimo utilizando una lista de nodos abiertos y cerrados. En cada iteración, se selecciona el nodo con el menor costo total (`f`) de la lista de nodos abiertos y se expande para evaluar sus nodos vecinos. Se calcula el costo acumulado desde el nodo inicial hasta cada vecino (`g`), así como el costo estimado hasta el nodo final (`h`). Luego, se actualiza el costo total (`f`) y se agrega el vecino a la lista de nodos abiertos si no está en la lista de nodos cerrados. Este proceso se repite hasta que se encuentra el nodo final o se recorren todos los nodos posibles.

Además del algoritmo principal, se proporcionan funciones para la manipulación de la cuadrícula, como la definición de obstáculos, puntos de inicio y destino, y la visualización del camino encontrado. También se incluyen funciones para la gestión de puntos de referencia intermedios, denominados "waypoints", que permiten al usuario definir múltiples destinos en el camino.

Por último, el concepto de montañas en la implementación del algoritmo A* se refiere a áreas de la cuadrícula que representan obstáculos de mayor altura, lo que implica un mayor costo para atravesarlas. En el código proporcionado, las montañas se distinguen de otros obstáculos mediante la propiedad `mountain` de los nodos en

la cuadrícula. Cuando el usuario marca un nodo como una montaña, se establece esta propiedad en `true`.

La funcionalidad relacionada con las montañas se lleva a cabo durante la expansión de los nodos vecinos de un nodo dado, se realiza una verificación adicional para determinar si un vecino es una montaña y, en ese caso, si la altura del avión es suficiente para superarla. Esto se hace evaluando la diferencia de alturas entre el nodo actual y el vecino que representa la montaña, comparándola con la altura del avión definida por el usuario. Si la altura del avión es menor que la altura de la montaña, se considera que el vecino es inaccesible y se omite en el proceso de búsqueda.