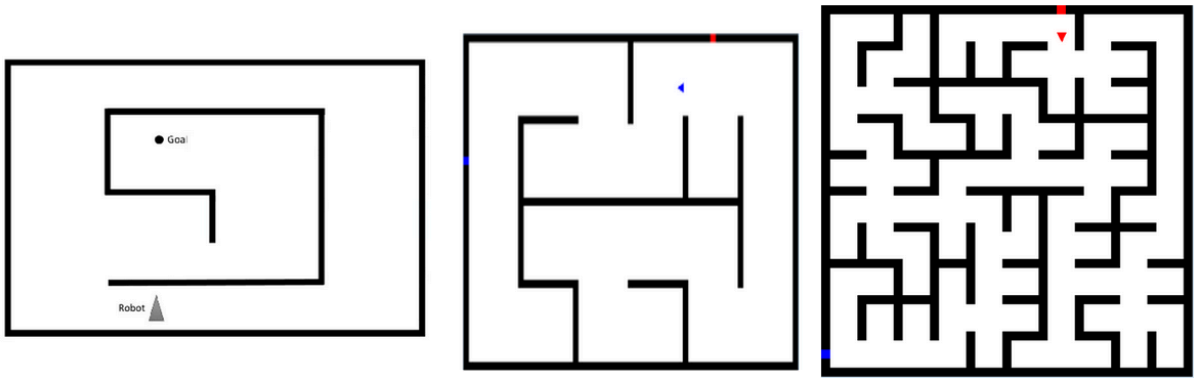
José Jezarel Sánchez Mijares
A01735226

1. Obtener el análisis de los posibles casos para cada uno de los algoritmos de evasión de obstáculos aplicados en los siguientes escenarios:



2. Argumentar en cada caso cuales son los algoritmos que garantizan que siempre el robot alcance el objetivo planteado
3. Generar un reporte de la comparativa de los algoritmos para cada escenario. Incluir los diagramas de todos los posibles casos de navegación reactiva

Tipos de algoritmos

Algoritmo Bug 0
El algoritmo Bug 0 es el más simple de los tres algoritmos, pero también es el menos eficiente. Para resolver el laberinto, el robot comienza en la posición inicial y sigue la pared derecha hasta encontrar una salida o llegar a un callejón sin salida. Si llega a un callejón sin salida, gira a la izquierda y sigue la pared izquierda hasta encontrar una salida o llegar a otro callejón sin salida. Este proceso se repite hasta que el robot encuentra la salida del laberinto.

Algoritmo Bug 1
El algoritmo Bug 1 es una mejora del algoritmo Bug 0. Para resolver el laberinto, el robot comienza en la posición inicial y sigue la pared derecha hasta encontrar una salida o llegar a un callejón sin salida. Si llega a un callejón sin salida, gira a la izquierda y sigue la pared izquierda hasta encontrar una salida o llegar a otro callejón sin salida. Sin embargo, si el robot llega a un callejón sin salida por segunda vez, gira a la derecha y sigue la pared derecha hasta encontrar una salida o llegar a un callejón sin salida. Este proceso se repite hasta que el robot encuentra la salida del laberinto.

Algoritmo Bug 2
El algoritmo Bug 2 es el más eficiente de los tres algoritmos. Para resolver el laberinto, el robot comienza en la posición inicial y sigue la pared derecha hasta encontrar una salida o llegar a un callejón sin salida. Si llega a un callejón sin salida, gira a la izquierda y sigue la pared izquierda hasta encontrar una salida o llegar a otro callejón sin salida. Sin embargo, si el robot llega a un callejón sin salida por segunda vez, gira a la derecha y sigue la pared derecha hasta encontrar una salida o llegar a una celda que ya ha visitado. Si el robot llega a una celda que ya ha visitado, gira a la izquierda y sigue la pared izquierda hasta encontrar una salida o llegar a otra celda que ya ha visitado. Este proceso se repite hasta que el robot encuentra la salida del laberinto.

Para el análisis de estos 3 casos es importante analizar las distintas características de los laberintos en cuestión, siendo que en todos los casos cuentan con estructuras cerradas y esto puede llegar a ser un

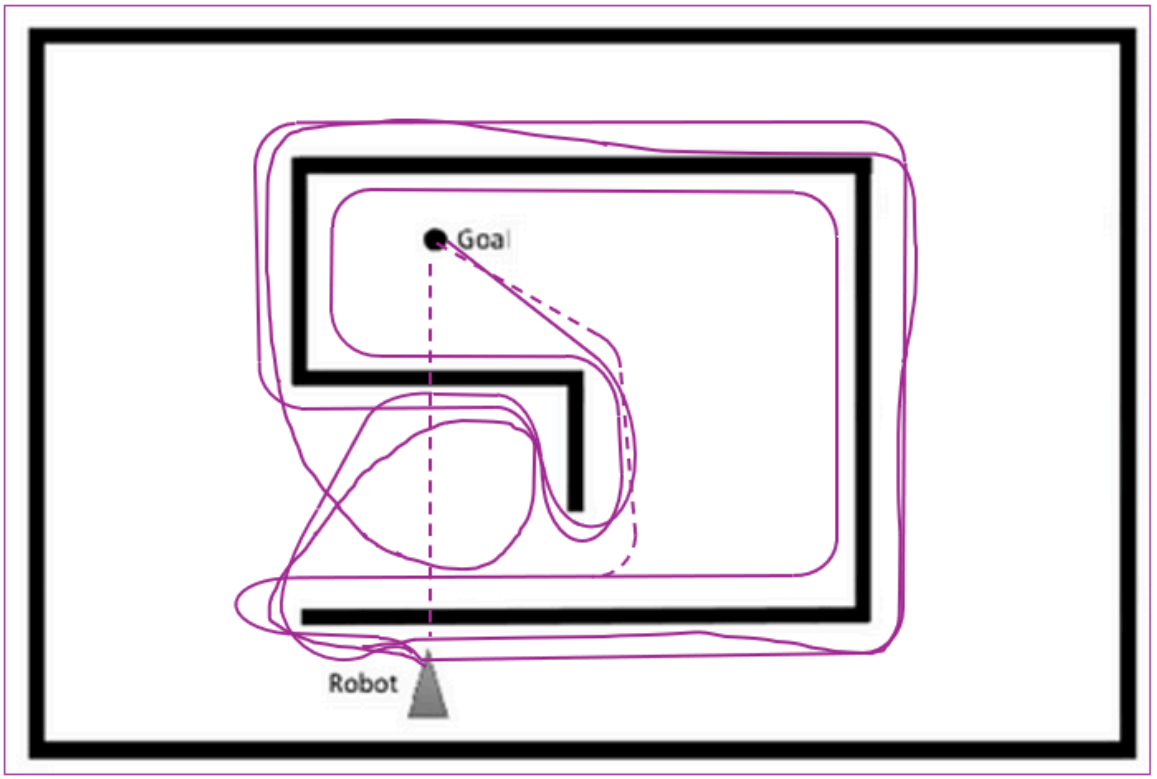
problema ya que puede llegarse a dar el caso en que existan redundancias al momento de tratar de completar el recorrido.

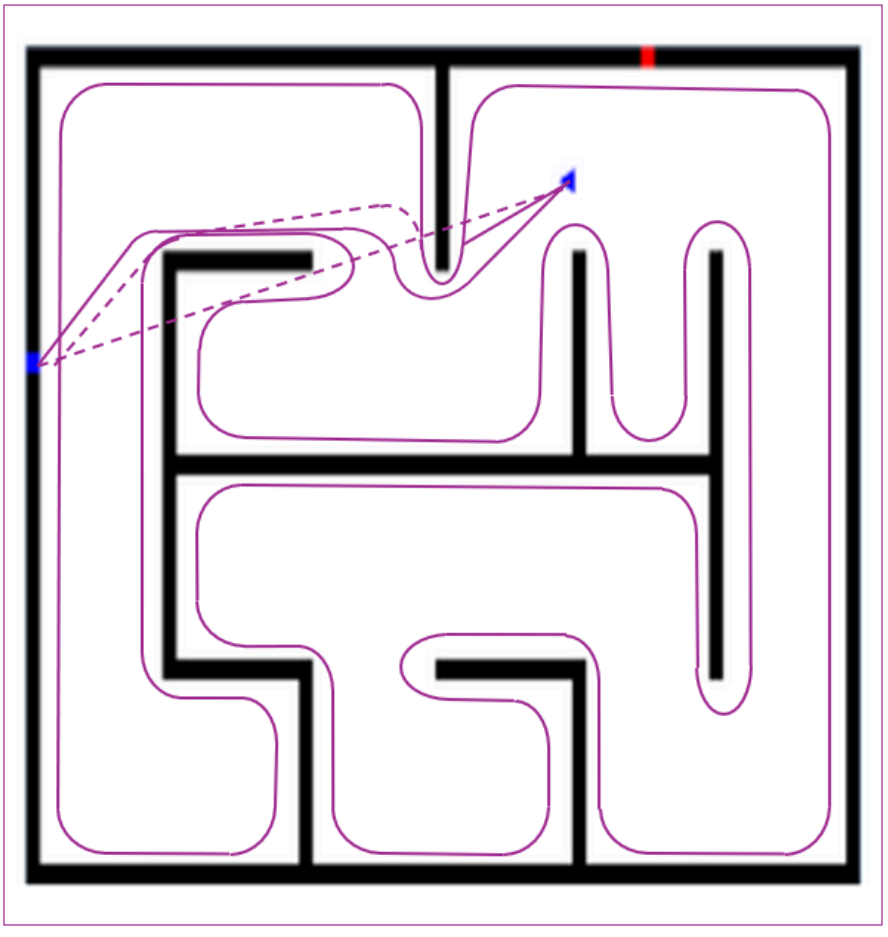
En el primer caso podemos ver que es el más simple entre los 3 casos, sin embargo cuenta con una complejidad elevada para los algoritmos. El primer algoritmo, el bug0 no podría completar el recorrido ya que este quedaría ciclado y no podría completar el recorrido. Por otra parte tanto el Bug 1 y el Bug 2 pueden resolver el caso ya que el estos sí podrían llegar a su goal debido a su algoritmo, sin embargo el bug1 tardaria más en realizar el recorrido

Para el segundo caso podemos visualizar que este laberinto es un poco más complejo ya que cuenta con más paredes, un punto de acceso y uno de salida, entonces conociendo los principios de funcionamiento de los algoritmos nos podemos dar cuenta que los algoritmo que lo pueden resolver son el bug1 y el bug 2 siendo que el algoritmo bug 1 es menos eficiente que el bug 2, sin embargo estos son los más adecuados para poder resolver este laberinto. En el caso de bug uno recorrería toda la figura ocasionando que utilicé muchísimo tiempo en poder en recorrer toda la estructura y así busque el mejor camino al goal, por otra parte el goal 2 es el mejor algoritmo ya que al trazar una línea recta a la meta permite el poder hacer todo el recorrido, sin embargo este puede llegar a tener unos problemas si llegase a encontrar un camino algo complejo podría llegar a hacer un tipo de ciclado pero en esta ocasión esto no ocurre.

En el último caso podemos encontrar un laberinto mucho más complejo, siendo que este cuenta con más paredes y por ende una forma más compleja de llegar a la meta. de todos los algoritmos el algoritmo Bug 2 es el más eficiente para resolver el laberint. Sin embargo, el algoritmo Bug 1 también es una buena opción, ya que es más simple de implementar y todavía puede encontrar el camino óptimo al laberinto en la mayoría de los casos. El algoritmo Bug 0 no es recomendable para este problema, ya que es el menos eficiente y es más probable que recorra un camino más largo que el camino óptimo. La eficiencia de cada algoritmo también dependerá del tamaño y la complejidad del laberinto. En general, los algoritmos más eficientes son más adecuados para laberintos grandes y complejos. También es importante tener en cuenta que los algoritmos de Bug no son perfectos. En algunos casos, pueden quedar atrapados en bucles infinitos o no encontrar la salida del laberinto. Por lo tanto, es importante utilizar estos algoritmos con precaución y tener un plan de respaldo en caso de que fallen.

José Jezarel Sánchez Mijares
A01735226





José Jezarel Sánchez Mijares
A01735226