

Entrega Parcial del Proyecto



Xavier Cifuentes Pérez 13316
Pablo de León Rodas 13227
Ana Fernanda Solares 13125
Jose Rosales 12576
Julio Isaac Rodriguez 14298
Julian Estuardo Escobedo 14684

Facultad de Ingeniería
Algoritmos y estructura de datos
Universidad del Valle de Guatemala

Algoritmos para salir de laberintos

- **Algoritmo de Azar**

Este algoritmo consiste en que el robot toma arbitrariamente la dirección de cada uno de los nodos que corresponden a las celdas que tiene más de una posible dirección, por lo que el robot tiene altas posibilidades de llegar a la meta, este algoritmo tiene varias ventajas como las de no se requiere mucha capacidad de procesamiento, pero consta de varias desventajas a la vez como la de ausencia total de inteligencia, no sabe en qué momento el robot ha llegado a la meta y tampoco se puede determinar cuánto tiempo se tardará el robot en terminar el laberinto.

- **Algoritmo de seguidor de paredes**

Este algoritmo se basa en el seguimiento de paredes, izquierda o derecha, al realizar este seguimiento invariablemente el robot encontrará la salida. El algoritmo se basa en el seguimiento de banderas, en las cuales se colocan 0 si no encuentra la pared que se eligió y un 1 si se encontró la pared seleccionada, si el robot lee una bandera 1 realizará una acción, y si lee una bandera 0 girará y buscará cambiar el valor de la bandera sea 1, este algoritmo tiene la ventaja de que no se necesitan componentes complejos y se puede realizar utilizando algunos sensores no tan complicados de utilizar.

- **Algoritmo de búsqueda por anchura en un grafo (BFS)**

Este algoritmo parte de un nodo inicial que será considerado la raíz del árbol que se irá formando conforme el robot vaya avanzando dentro del laberinto, luego ve las posibles salidas adyacentes y las agrega en una cola, como la prioridad de la cola es FIFO (primero en entrar, primero en salir), los siguientes nodos a evaluar serán los adyacentes a los que ya estaban insertados. Así conforme el robot vaya avanzando en el laberinto va determinando si es un lugar en el cual puede seguir avanzando o detecta que no puede seguir, al determinar que si puede seguir avanza, y sino retrocede al último lugar que analizo.

- **Algoritmo seleccionado**

Se seleccionó el algoritmo de búsqueda por anchura en un grafo (Tremaux), ya que se puede implementar estructuras de datos que tienen como base las pilas, además es el algoritmo que es más eficiente de los que encontramos en cuanto a que el robot tiene la capacidad de guardar el último lugar en el que está y decidir si puede avanzar o retroceder para volver a analizar el lugar y así poder encontrar una salida.

Estructura de datos a utilizar

Se usará una estructura de datos lineal estática con almacenamiento contiguo, pilas. Se quiere utilizar pilas porque los elementos solo pueden ser removidos o añadidos por un solo extremo, trabaja en LIFO (Last in - First Out). Además podemos tener operaciones básicas como realizar push (insertar elemento), pop (remover elemento), se puede indicar si la pila está vacía o si ya no se pueden seguir insertando elementos. En el robot se utilizarán tres sensores para determinar si hay pared o una posible salida, si es pared enviará un 0 y si es salida enviará un 1, si en una parte del laberinto no encuentra salida, retrocederá al lugar en donde encontró la última salida. Así recorrerá el laberinto hasta encontrar una salida.

Diagrama de flujo

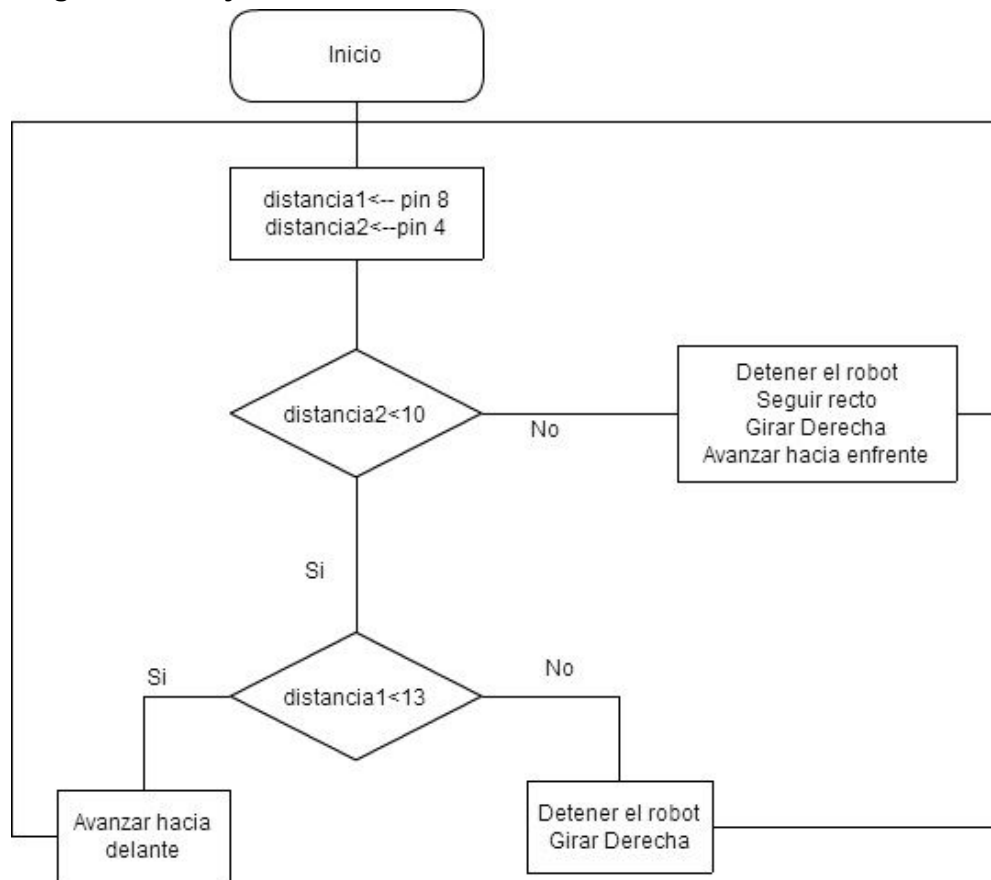


Diagrama de clases

