

Proyecto 1: Un robot sale de un laberinto

1. Algoritmos existentes para salir de un laberinto

a. Algoritmo de Trémaux

Inventado por ingeniero francés Charles Trémaux. Para que el algoritmo funcione se debe poder marcar el camino que se está tomando. El algoritmo dice lo siguiente:

- No tomar el mismo camino más de una vez
- Si el cruce es nuevo tomar cualquier dirección
- Si el camino nuevo llega a un camino viejo o a un callejón sin salida regresar al inicio
- Si un camino viejo lleva a un cruce ya tomado, elija un nuevo camino, si ninguno es nuevo elegir cualquiera

b. Algoritmo de la mano derecha

Este algoritmo es bastante simple y útil para laberintos cuyo objetivo es salir y no llegar al centro. Básicamente se trata de seguir siempre la derecha, como si se pegara una mano a la pared derecha y se siguiera todo el recorrido sin despegar la mano. Funciona de la misma manera para la mano izquierda.

c. Algoritmo de Pledge

Este es similar al de la mano derecha pero logra resolver el problema de encontrar la solución en el medio. Se escoge una dirección arbitraria inicial. Cuando llega a un obstáculo sigue el algoritmo de la mano derecha pero cuenta los giros. Cada giro a la izquierda es un -1 y cada giro a la derecha es un +1. Cuando el contador regrese a cero, significa que ya atravesó el obstáculo y puede seguir en su dirección original.

d. Random Mouse

Este es una solución trivial, ya que la lógica que sigue este algoritmo es en cada cruce tomar una dirección al azar hasta que salga del laberinto. Eventualmente encontrará el camino correcto pero si el laberinto es muy largo, se tomará mucho tiempo en salir.

2. Justificación del algoritmo a utilizar

Escogimos utilizar el algoritmo de la mano derecha. La razón de nuestra decisión es porque el de Trémaux es muy difícil de implementar en un robot, ya que es necesario marcar los cruces que ha tomado y de todas maneras el tiempo que requiere para resolver un laberinto puede ser muy tardado. El de Random Mouse es el menos eficiente de todos, por lo que fue descartado. El de Pledge hubiera servido, ya que es muy similar al que escogimos, pero el hecho de tener que contar cada vez que gire a la derecha o a la izquierda puede generar errores en el robot, ya que desconocemos la capacidad del mismo así como el nivel de dificultad de la programación.

3. Pseudocódigo/Diagrama de flujo

Inicio

```
Mientras (robot no ha salido) {  
    If (no hay pared a la derecha) {  
        girar a la derecha y avanzar }  
    else if (no hay pared al frente) {  
        avanzar al frente }  
    else { girar a la izquierda }  
}  
Fin
```

Referencias:

Wikipedia. 2016. "Maze solving algorithm".

https://en.wikipedia.org/wiki/Maze_solving_algorithm [con acceso el 01/08/2016].