Universidad Del Valle de Guatemala
Algoritmos y estructura de datos
Catedrático: Douglas Barrios

Proyecto No. 1: Laberinto

Jorge Azmitia 15202
Jennifer Barillas 15307
Carlos Calderón 15219

¿Que algoritmos existen para salir de un laberinto ?

Algoritmo de Tremaux

Inventado por el ingeniero Charles Tremaux. el cual sugiere la siguiente serie de pasos:

1. No siga el mismo camino dos veces.
2. Si llega a un cruce nuevo, no importa qué camino siga.
3. Si un camino nuevo lo lleva a un cruce viejo, o a un callejón sin salida, retroceda hasta la entrada del camino.
4. Si un camino viejo lo lleva a un cruce viejo, tome un camino nuevo, y si no lo hay, tome cualquiera.

Algoritmo random

Este es un método trivial, el cual simplemente depende de seguir el camino y al momento de tener la opción de girar hacia dos diferentes lugares, toma una decisión aleatoria, eventualmente se logra salir del laberinto, sin embargo puede llegar a ser extremadamente lento

Algoritmo de "seguir la pared"

También conocido como la regla de la mano izquierda o derecha. este algoritmo consiste en algo tan sencillo como que tanto la entrada como la salida del laberinto están conectadas. por lo tanto, si la persona que está dentro del laberinto, recorre este manteniendo la mano pegada a una de las dos paredes iniciales, sea cual sea. y sigue todo el laberinto así, eventualmente encontrará la salida, y de no existir ninguna salida regresará al punto de partida.

Algoritmo de pledge

Nombrado así por John Pledge. este algoritmo consiste en tomar de forma arbitraria una dirección por la cual ir. y que siempre gire hacia el mismo lado. en este algoritmo se tiene que ir tomando en cuenta los ángulos o giros que se dan al encontrar un obstáculo y girar en la dirección elegida. con esto se sabe si se dio una vuelta a una parte del laberinto, y si se regresa al giro anterior teniendo en cuenta que ya dio N vueltas y esta viendo hacia la dirección inicial, entonces ignora el giro y sigue hacia delante hasta encontrar el siguiente y eventualmente la salida. esto puede ser útil si en algún momento se encuentra uno con un laberinto segmentado. el cual podría solucionarse con el algoritmo anterior a menos que se le coloque como inicio en alguna parte central del laberinto y que esta no esté pegada a los bordes del laberinto.

Algoritmo backtracking

Este algoritmo se centra en el personaje dentro del laberinto, por lo general es rápido. Lo que hace este algoritmo es ver si ya se encontró con una pared o en un área ya conocida, regresa fracaso, asimismo si encuentra una salida retorna victoria. Sino, trata de forma recursiva de moverse en cuatro direcciones, (N,S,O,E). Cabe resaltar que este algoritmo, utiliza una pila con un espacio de memoria igual al del laberinto. Este dibuja una línea cuando prueba una dirección y lo borra cuando regresa fracaso.

Algoritmo a utilizar

Luego de estudiar las posibilidades, se vio que el algoritmo de Tremaux y el recursivo Backtracker tienen un garantía de solución bastante alta. Así también, se observó que tanto el algoritmo de Pledge como el de Seguidor de pared son aunque relativamente más simples que los ya mencionados, y por lo general son rápidos, aunque su garantía de solución es un poco más baja que los ya mencionados.

Así pues, se enfocó en el algoritmo de seguir a la pared con la mano derecha y con la izquierda , y se probó con el algoritmo de backtracking. Estas pruebas fueron hechas en c++, con distintos laberintos. Se observó que el backtracking efectivamente consume más recursos del sistema y depende la complejidad del algoritmo que tanto consumo puede llegar a tener. Por otra parte, el algoritmo de seguir a la pared aunque consume poco y es rápido, puede llegar a ser muy ineficiente pues lleva a cabo muchos más movimientos, aún en laberintos simples. Después de estudiar ambos, se decidió utilizar el de seguir a la pared.

Lo anterior debido a que si bien es cierto, el backtracking garantiza solución, siempre y cuando el laberinto la tenga. Este consumió demasiada memoria en las pruebas que se utilizaron. Para un laberinto pequeño fue eficiente, pero para uno de alrededor del doble de tamaño, el consumo de memoria que hizo fue abismal. Por otra parte, el algoritmo de seguir a la pared para ambos laberintos ejecuto muchos movimientos, pero su consumo de recursos no es tan brutal.

Inicialmente, la estructura de datos que se presentan en las pruebas son arreglos dinamicos para guardar el camino recorrido en el algoritmo backtracking.

Pseudocódigo:

```
Inicio
Mientras( robot no sale)
        Si (no hay pared a la derecha){
                girar a la derecha y avanzar}
        Si (no hay pared enfrente){
                avanzar al frente}
        Sino { girar a la izquierda}
}
Fin
```

Bibliografía

- https://www.cs.bu.edu/teaching/alg/maze/
- Turtle Geometry:computer as a medium for exploring mathematics
- MIT AI, Uses of technology to enhance education
- Edouard lucas: recreations Mathematiques V1, 1882

- https://en.wikipedia.org/wiki/Maze_solving_algorithm