Universidad del Valle de Guatemala Luis Diego Sierra Cordón, Carnet: 131074

Colegio Universitario Salvador Antonio Recinos Diaz, Carnet: 15001

Algoritmos y estructuras de datos Carlos Alberto Solorzano Pineda, Carnet:

2 de Agosto de 2016 Jose Giovani Tzoc Velásquez, Carnet: 13322

**Investigación**

**Algoritmos existentes para resolver un laberinto:**

Algoritmos de Tremaux:

* El algoritmo de Tremaux es un algoritmo inventado por Charles Tremaux para hallar la salida de un laberinto. Lo primero que se hace es marcar el camino y se siguen los siguientes pasos:
* No seguir el mismo camino 2 veces.
* Al llegar a un cruce nuevo, no importa que camino siga.
* Si un camino nuevo lo lleva a un cruce viejo, retrocede hasta el camino viejo.
* Si un camino viejo lo lleva a un cruce viejo, debe tomar un camino nuevo, sino hay caminos nuevos debe tomar cualquiera.

(Kim, J. 2014.)

Wall Follower:

* Es un algoritmo el cual consiste en seguir la pared derecha o izquierda hasta llegar a la salida. Esto funciona solamente si el laberinto está completamente conectado. Este algoritmo es el mejor algoritmo conocido para atravesar laberintos. Si las paredes no están conectadas entre sí el método no funciona ya que no podría seguir una pared hasta llegar al final.

(Kim, J. 2014.)

Algoritmo del camino más corto:

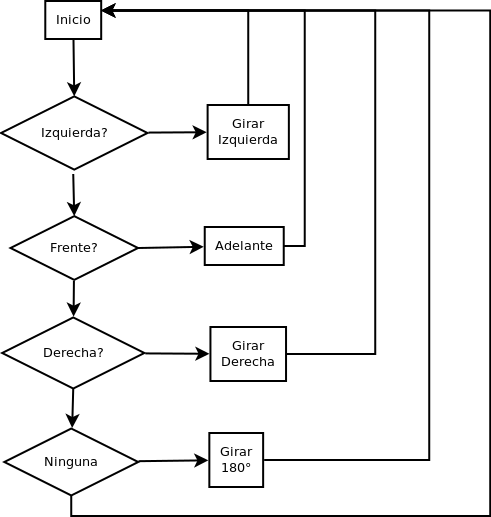
* Los laberintos tienen varias soluciones, hay varios algoritmos para encontrar el camino más corto la mayoría de ellos procedentes de la teoría de grafos. Un algoritmo posible de encontrar el camino más corto mediante la anchura, mientras que el otro el algoritmo A\* con una teoría heurística.

(Johnson, D. 2009.)

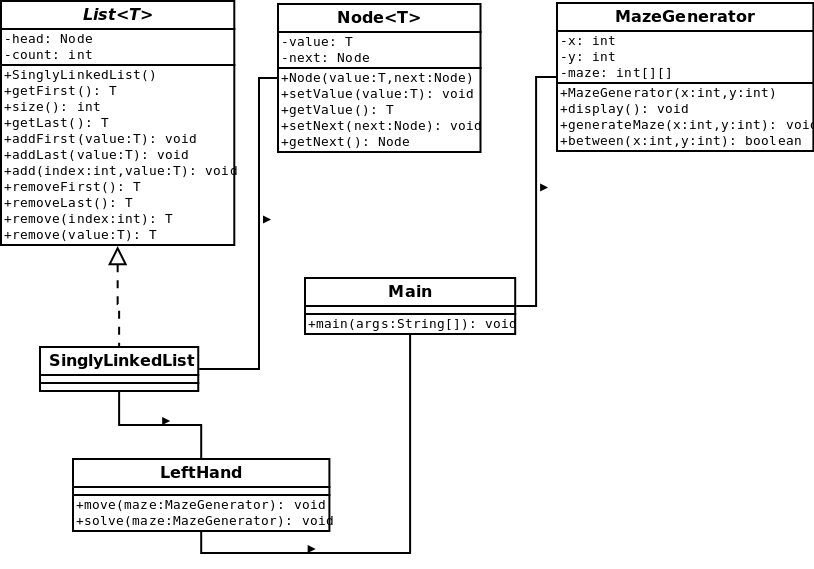
**Algoritmo a utilizar:**

El algoritmo que se escogió fue el de Wall follower debido a que dicho algoritmo es el más efectivo a la hora de salir de un laberinto. La metodología que se pensó para aplicar dicho algoritmo fue el de *left-hand-rule* el cual consiste en que el robot siempre va a seguir la pared izquierda para lograr llegar a su objetivo final.

Se eligió el algoritmo Wall-follower debido a que este algoritmo, una vez todas las paredes estén conectadas entre sí, encuentra siempre la salida al laberinto. Este consiste en moverse pegado hacia algún lado (derecho o izquierdo), y cuando se termine la pared, decidir hacia dónde continuar. Para evitar loops, se decidió utilizar Listas simplemente enlazadas como estructura de datos para almacenar el recorrido dentro del laberinto (sentido del movimiento).

**Diagrama de flujo:**

**UML:**



**Referencias:**

* Kim, J. 2014. *Robot Intelligence Technology and applications 2*. Springer. Korea.
* Johnson, D. 2009. *Dimacs Implementation Problem.* Camil. USA.