Universidad del Valle de Guatemala. Algoritmos y Estructuras de Datos. Sección 20
German García 15008
Antonio Ixtecoc 15582
Aldo Aguilar Nadalini 15170
Esteban Avalos 15059
Luis Nájera 15581

Primera Fase Proyecto

- La investigación de los algoritmos existentes para resolver la situación que se presenta.
- La explicación de la selección del algoritmo a utilizar en el proyecto.
- Explique la estructura de datos a utilizar, y diseñe el diagrama de clases que utilizará.
- El pseudocódigo o diagrama de flujo del algoritmo.

Investigación

Backtracking:

El algoritmo Backtracking es utilizado para la resolución de problemas donde se requiere de una serie de decisiones. En el caso de laberintos el algoritmo funciona de tal manera que genera todas las secuencias necesarias hasta encontrar la correcta, almacenando las secuencias erróneas para no generarlas nuevamente.

En este caso el orden en que se forman las secuencias no es importante, pues, de ser incorrecta la secuencia el cómputo retrocede y comienza de nuevo a partir del punto de decisión anterior. Este método utiliza funciones recursivas las cuales asignan valores específicos a puntos donde existe una posible solución.

Right hand:

El algoritmo de right hand es un algoritmo basado en la repetición de la instrucción de revisar si el franco derecho se encuentra libre y si esta libre el robot se dirige hacia esa parte del laberinto. Cómo se implementa el algoritmo de forma cíclica dado que siempre se realizará la misma instrucción.

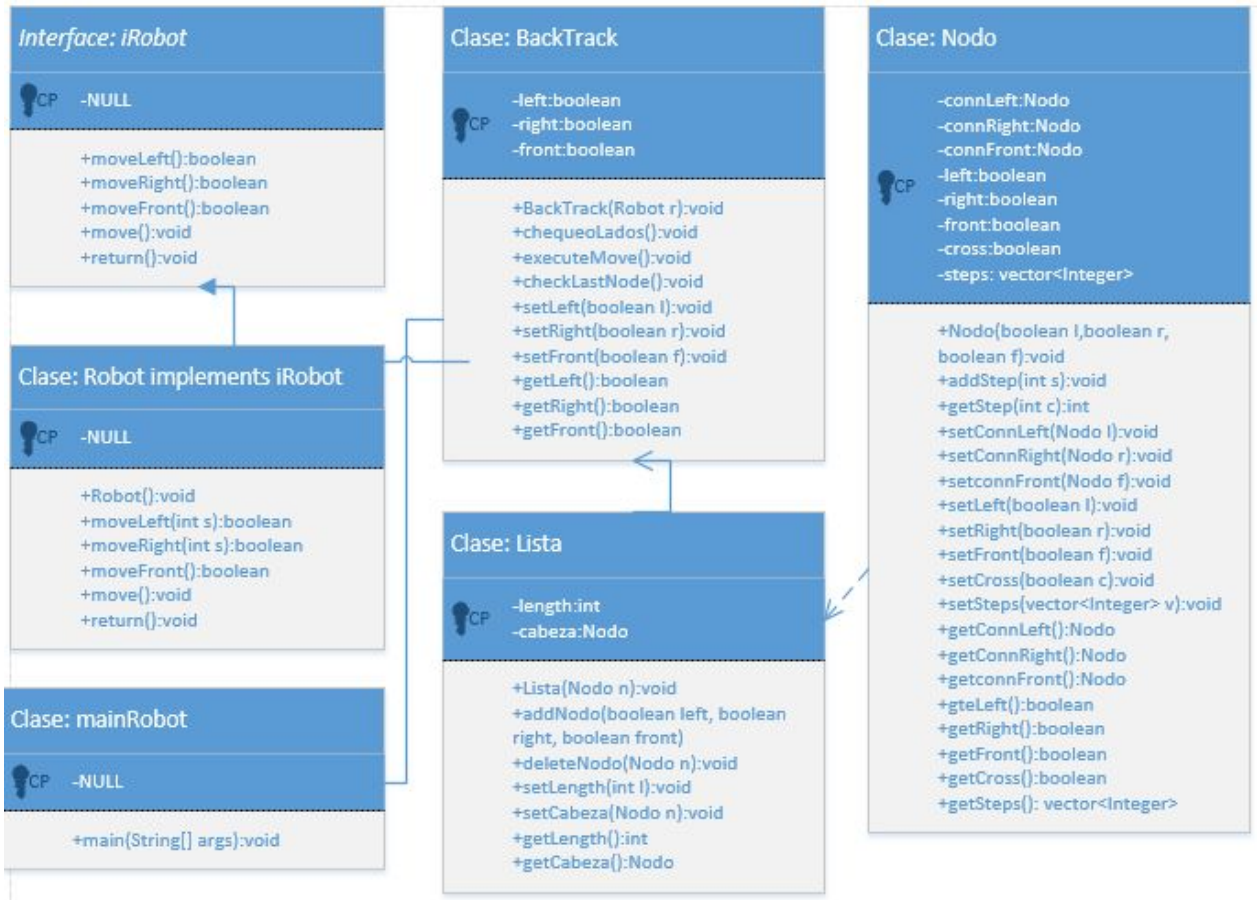
¿Por qué Elegimos el Algoritmo de Backtracking?

A nuestro parecer, el método de backtracking es el más eficiente de los que tenemos disponibles. En cuanto a opciones, hay una infinidad de maneras de salir de un laberinto, pero entre todas esta opción fue la que más nos pareció debido a que este da la oportunidad de reconocer los pasos ya recorridos y los guarda en una memoria, la cual se usa para idear la mejor estrategia de salir del laberinto. Esto lo implementamos debido a que podemos relacionar los pasos con nodos de listas, y así poder hacer eficiente la salida del robot, mientras ponemos o quitamos pasos a seguir de la "pila" de pasos.

Estructura de Datos

En cuanto a la estructura de datos a implementar en el proyecto, se utilizará una lista de nodos dinámica estructurada por el grupo, en la cual se almacenarán los datos del recorrido que realizará el robot, así como el proceso de toma de decisiones de caminos a seguir en el laberinto y recursión de los pasos para salir de callejones sin salida. Dentro de los nodos también se implementarán vectores para almacenar valores de enteros simples.

Diagrama de Clases



Pseudocódigo

Inicialización de clase robot, inicialización de clase BackTrack y Lista, añadir primer nodo cabeza. Mientras sea verdadero: realizar chequeo de lado izquierdo, derecho y frontal, retornar estados de lados, actualizar atributos de clase BackTrack. Analizar estados de clase BackTrack. Si estado frontal es verdadero, aumentar contador de nodo actual, ejecutar un paso hacia adelante. En cambio, si existe cruce, añadir nodo nuevo a lista, actualizar nodo actual e identificarlo con el tipo de cruce (izquierda o derecha). Si existen bifurcación, añadir Nodo Especial, enlazar nodos de cruce. Tomar primer nodo cruce como nodo actual. Regresar a inicio de ciclo. Si ningun lado libre, rotar 180° grados. Reducir contador por cada paso dado hasta llegar a nodo de cruce. Nodo cruce es nuevo nodo actual. Si nodo cruce es derecha, girar izquierda. Si es nodo cruce izquierda, girar derecha. Seguir reduciendo contador de nodo actual. Si llega a Nodo Especial, cambiar de nodo opción y eliminar nodo opción utilizado. En cambio, si ya no existe nodo opción, regresar hasta siguiente Nodo Especial. Regresar a inicio de ciclo. Break.

Referencias:

De Casas, Carlos. El esquema algorítmico del Backtracking. http://www.it-docs.net/ddata/3619.pdf [Consultado: 02/08/16]

Diaz, Ariel. Algoritmo de backtrackingrecursivo y no recursivo para la resolución de unlaberinto y su aplicación enSDL. http://www.academia.edu/4607412/Algoritmo_de_Backtracking_Rec

ursivo_y_no_Recursivo_para_la_Resoluci%C3%B3n_de_un_Laberinto_y_su_Aplicaci%C3%B3n_en_SDL [Consultado: 02/08/16]