
SISTEMA DE CONTROL PARA CHAPIN WARRIORS S.A.

202004763 – Cristian Noé Axpuc Aspuc

Resumen

Chapin Warrior ha desarrollado equipos para salvar civiles y recursos de una ciudad, estos funcionan mediante comandos de configuración en un archivo con extensión xml. El siguiente programa cumple la labor de extraer las instrucciones del archivo con extensión xml e implementarlas en las distintas ciudades ubicadas en el archivo xml y programar al robot que será enviado a dicha tarea con la finalidad de que este encuentre un camino capaz de completar la misión de rescate de las distintas unidades encontradas por el autómata Chapin Eyes en la ciudad seleccionada.

Palabras clave

XML, Lista simple, Path Finding, Graphviz, clasificación de puntos.

Abstract

Chapin Warrior has developed teams to save civilians and resources of a city, these work through configuration commands in a file with an xml extension. The following program performs the task of extracting the instructions from the file with an xml extension and implementing them in the different cities located in the xml file and programming the robot that will be sent to said task in order for it to find a way to complete the mission. of rescue of the different units found by the automaton Chapin Eyes in the selected city.

Keywords

XML, Simple List, Pathfinding, Graphviz, Point Sorting.

Introducción

El siguiente programa fue creado a partir de la implementación de una TDA propia llamada Lista() la cual tiene la funcionalidad de una Lista Simple Enlazada para guardar en ella los datos extraídos de un archivo con extensión XML el cual contiene distintas instrucciones para controlar al robot seleccionado para la misión dentro de la ciudad también extraída del mismo archivo.

Desarrollo del tema

Clase: Node():

Esta clase fue utilizada para implementar nodos con atributos data y siguiente, de esta forma se pudo implementar la lista enlazada dado que mediante el ciclo

```
For i in range(0, self.cantidad_datos():  
    Self.root.siguiente
```

Donde self.root es el atributo nodo en la lista, se utiliza el ciclo for, se delimita el rango por range(0, self.cantidad_datos()) y se recorre por i.

Definición de Nodo:

En estructuras de datos dinámicas un nodo es un registro que contiene un dato de interés y al menos un puntero para referenciar (apuntar) a otro nodo. Si la estructura tiene solo un puntero, la única estructura que se puede construir con él es una lista, y si el nodo tiene más de un puntero se pueden construir estructuras más complejas como árboles o grafos.

Clase: Lista():

Esta es la clase donde mayor cantidad de procesos se llevan a cabo, esta clase construye una lista enlazada simple en la cual se almacena toda la información del archivo xml y además en ellas se llevan a cabo procesos secundarios llamados por la Clase: MAIN().

Definición de Lista simple:

En ciencias de la computación, una lista enlazada es una de las estructuras de datos fundamentales, y puede ser usada para implementar otras estructuras de

datos. Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior. El principal beneficio de las listas enlazadas respecto a los vectores convencionales es que el orden de los elementos enlazados puede ser diferente al orden de almacenamiento en la memoria o el disco, permitiendo que el orden de recorrido de la lista sea diferente al de almacenamiento.

Una lista enlazada es un tipo de dato autorreferenciado porque contienen un puntero o enlace (en inglés link, del mismo significado) a otro dato del mismo tipo. Las listas enlazadas permiten inserciones y eliminación de nodos en cualquier punto de la lista en tiempo constante (suponiendo que dicho punto está previamente identificado o localizado), pero no permiten un acceso aleatorio. Existen diferentes tipos de listas enlazadas: listas enlazadas simples, listas doblemente enlazadas, listas enlazadas circulares y listas enlazadas doblemente circulares.

Clase: graficar():

Esta clase es utilizada con el objetivo de mostrar gráficamente la ciudad seleccionada, recibe como parámetros el código del patrón de la ciudad a graficar y la descripción de la misión. Utiliza como motor la librería graphviz la cual se basa en un sistema de nodos con el objetivo de generar una gráfica enlazada

·
Graphviz:

Instalación:

graphviz proporciona una interfaz simple de Python puro para el software de dibujo de gráficos Graphviz. Se ejecuta bajo Python 3.6+. Para instalarlo con pip, ejecuta lo siguiente:

```
$ pip install graphviz
```

Para una instalación en todo el sistema, esto generalmente requiere acceso de administrador. Para

una instalación aislada, puede ejecutar lo mismo dentro de a venvo virtualenv.

La única dependencia es una instalación funcional de Graphviz (página de descarga, versiones archivadas, procedimiento de instalación para Windows).

Después de instalar Graphviz, asegúrese de que su bin/subdirectorio que contiene el dot comando de diseño para representar descripciones de gráficos esté en sus sistemas PATH (a veces lo hace el instalador; configuración PATH en Linux, Mac y Windows): en la línea de comandos, debe imprimir la versión de su instalación de Graphviz.dot -V

Uso:

El paquete graphviz proporciona dos clases principales: graphviz.Graph y graphviz.Digraph. Crean descripciones de gráficos en el lenguaje DOT para gráficos dirigidos y no dirigidos, respectivamente. Tienen la misma API.

Lenguaje Dot:

La siguiente es una gramática abstracta que define el lenguaje DOT. Los terminales se muestran en negrita y los no terminales en cursiva. Los caracteres literales se dan entre comillas simples. Paréntesis (e) indicar agrupación cuando sea necesario. Corchetes [y] encierre elementos opcionales. Las barras verticales |separan las alternativas.

Las palabras clave nodo, borde, gráfico, dígrafo, subgráfico y estricto son independientes de mayúsculas y minúsculas. Tenga en cuenta también que los valores de puntos de la brújula permitidos no son palabras clave, por lo que estas cadenas se pueden usar en otros lugares como identificadores ordinarios y, a la inversa, el analizador aceptará cualquier identificador.

Una identificación es una de las siguientes:

Cualquier cadena de [a-zA-Z\200-\377]caracteres alfabéticos (), guiones bajos ('_') o dígitos ([0-9]), que no comience con un dígito; a numeral [-]?([0-9]+ | [0-9]+([0-9]*)?); cualquier cadena entre comillas

dobles ("...") que posiblemente contenga comillas escapadas ("\\"); una cadena HTML (<...>).

Una ID es solo una cadena; la falta de caracteres de comillas en las dos primeras formas es solo por simplicidad. No hay diferencia semántica entre abc_2y "abc_2", o entre 2.34y "2.34". Obviamente, para usar una palabra clave como ID, debe estar entre comillas. Tenga en cuenta que, en las cadenas HTML, los corchetes angulares deben aparecer en pares coincidentes, y se permiten nuevas líneas y otros caracteres de espacio en blanco de formato. Además, el contenido debe ser XML legal, por lo que las secuencias de escape XML especiales para ", &, < y > pueden ser necesarias para incrustar estos caracteres en valores de atributo o texto sin formato. Como ID, una cadena HTML puede ser cualquier cadena XML legal. Sin embargo, si se usa como un atributo de etiqueta, se interpreta de manera especial y debe seguir la sintaxis de las etiquetas similares a HTML.

Funciones PathFindingRescue() y PathFindigRecursos():

Son las clases encargadas de generar un camino para los robots estas clases se basan en el algoritmo a-estrella, estas buscan un punto de inicio a lo largo del mapa de la ciudad y el punto de salida.

Conclusiones

Las listas simples implementadas gracias a la ayuda de la definición de nodo sirven para guardar datos de manera ordenada dentro de estos. Además de ser la base para las demás estructuras de datos, las listas simples son la mejor herramienta de un programador para resolver problemas y mejorar su abstaccion.

Referencias bibliográficas

Graphviz, (27 de enero del 2022). Dot Lenguaje

<https://www.graphviz.org/doc/info/lang.html>

National Institute of Standards and Technology
(16 de Agosto del 2004). Definition of a linked
list.

<https://xlinux.nist.gov/dads/HTML/linkedList.htm>
!

DIAGRAMA DE CLASES:

