
CAMINO MAS CORTO PARA EL RECORRIDO DEL ROBOT R2E2

202004734 – Justin Josue Aguirre Román

Resumen

Se realizó un software en el cual se resuelven el problema para encontrar camino que el robot R2E2 deberá recorrer para consumir menor cantidad de gasolina analizando distintos terrenos. Empleando estructuras de datos dinámicas como: listas enlazadas y pilas especiales; y el paradigma de programación orientada a objetos (POO).

Consiste en la lectura de un archivo de entrada de tipo XML, archivo que contiene la información, que al ser ingresado se interpretará su información creando las estructuras necesarias para cada terreno inscrito en el, procesándolo y recorriendo su información para crear la ruta optima. Ruta que será exportada en forma de un archivo XML.

Las listas facilitaron el acceso a la información de los terrenos, así como la adaptabilidad a las dimensiones del terreno.

Palabras clave

Estructura de Datos, Clases, Software, Memoria, Algoritmos

Abstract

A software was made in which the problem was solved to find the path that the R2E2 robot must travel to consume less gasoline by analyzing different terrain. Using dynamic data structures such as: linked lists and special stacks; and the object-oriented programming (OOP) paradigm.

It consists of reading an XML-type input file, a file that contains the information, which when entered, its information will be interpreted creating the necessary structures for each land registered in it, processing it and going through its information to create the optimal route. Path that will be exported in the form of an XML file

The lists facilitated access to information about the land, as well as the adaptability to the size of the land.

Keywords

Data Structure, Classes, Software, Memory, Algorithms

Introducción

Se implementaron las estructuras de datos dinámicas para el desarrollo del software para así contar con un almacenamiento que sea capaz de procesar una gran cantidad de terrenos.

El paradigma de programación orientada a objetos, permite la creación de objetos, cruciales para el desarrollo de este software, separando cada porción del terreno con sus respectivas características, siendo cada interpretación de mayor facilidad.

Explicación del programa

Al ejecutarse el software, se mostrará un menú con las distintas opciones con las que cuenta: Leer Archivo, Procesar Archivo, Imprimir Archivo, Mostrar Datos del estudiante, Generar Graficas de terreno y Salir. Cada una de estas opciones serán explicadas a continuación.

El funcionamiento principal del software consiste la lectura y análisis de archivos de tipo XML, guardando su información agrupada en terrenos y las posiciones de cada terreno.

XML es el acrónimo de Extensible Markup Language, es decir, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos. Simplificando los conceptos expuestos podemos comenzar con que:

El lenguaje de marcado es un conjunto de códigos que se pueden aplicar en el análisis de datos o la lectura de textos creados por computadoras o personas. El lenguaje XML proporciona una plataforma para definir elementos para crear un formato y generar un lenguaje personalizado.

La programación informática considera que un nodo es cada uno de los elementos de una lista enlazada, un árbol o un grafo en una estructura de datos. Cada nodo tiene sus propias características y cuenta con varios campos; al menos uno de éstos debe funcionar como punto de referencia para otro nodo.

Una lista enlazada es una estructura de datos que puede utilizarse para la implementación de nuevas estructuras (tales como las colas, las pilas y sus

derivados) y está formada por una serie de nodos que almacenan, además de la información deseada, un enlace, un puntero o una referencia al nodo que lo precede, al posterior, o bien uno a cada uno. La ventaja fundamental de una lista enlazada en comparación con un vector convencional es que sus elementos no presentan un orden rígido ni relacionado con el que tuvieron al momento de ser almacenados, sino que éste depende del enlace que posee cada nodo, y puede ser modificado cuando así se desee.

Los archivos XML están conformados por la información tanto del terreno: nombre, dimensiones, posición de inicio y posición de salida; como de sus posiciones: su valor en Y, valor en X y la gasolina necesaria para atravesar ese posicionamiento como se muestra en la Figura 1.

La lectura de los archivos de entrada se utilizó la librería “etree”, importándola internamente desde Python, que trabaja este tipo de archivos con la estructura de árbol de nodos, es decir, trabaja los archivos con ramificaciones y distintas propiedades, cada ramificación equivale a un nivel de profundidad, para acceder a niveles mas profundos se agregaran un par de corchetes con el identificador de hijo que queramos acceder.

La biblioteca ElementTree incluye herramientas para analizar XML usando APIs basadas en eventos y documentos, buscando documentos analizados con expresiones XPath, y creando nuevos o modificando documentos existentes.

Para el análisis de los datos y su almacenamiento se optó por la creación de nodos, contando con dos tipos distintos de nodos, conectados por medio de listas enlazadas de n enlaces. la lista principal se utilizaría para guardar el primer tipo de nodos, siendo estos los nodos encargados de guardar la información de cada terreno (únicamente información propia del terreno como sus dimensiones o su nombre, excluyendo la información de las posiciones que lo componen). Esta lista solo cuenta con un enlace que siempre apuntara al siguiente nodo añadido a la lista. Los nodos ubicados en esta lista fueron encontrados con la instrucción cíclica “for” el cual en Python puede actuar implementando una especie de instrucción de

toma de decisión dentro de su misma sintaxis. En este caso cada vez que la lectura de la estructura de árbol se encuentre con la etiqueta terreno creará un nuevo objeto con las características en el XML y unas extras que manejarán el combustible utilizado en dicho terreno, además de contar con una verificación de si dicho terreno ha sido analizado con anterioridad.

De estos nacería la construcción de una nueva lista enlazada de n enlaces que simularía gráficamente una matriz, debido a que cada nodo se colocaría según como lo indiquen sus coordenadas X y Y. Estos enlaces pueden tener hasta un total de 4 enlaces, simulando cada uno de los puntos cardinales, y como mínimo pueden contar con 2 enlaces: esto al estar ubicados en alguno de los límites de las dimensiones del terreno. Estos nodos poseen propiedades brindadas por el XML, sus posibles enlaces y además cuentan con un Id y una verificación que confirmara si dicha posición es parte del recorrido óptimo o no.

Realizará el proceso de creación de estas estructuras hasta terminar de leer a plenitud el archivo XML. Luego de esto volverá al menú principal para solicitar el próximo proceso deseado.

El procesar el archivo se refiere al cálculo del camino a recorrer para un gasto de gasolina mínimo. Primero solicitará el nombre del terreno que deseamos procesar, luego comenzará a buscarlo a través de la lista enlazada principal con los punteros de sus nodos. Al encontrar el terreno solicitado precede a posicionarse en el nodo con coordenadas (1,1) para así buscar, por medio de estructuras cíclicas “while”, el nodo que coincida con las coordenadas de inicio requeridas por el terreno, moviéndose a cada nodo con los enlaces que el nodo en el que se encuentre posea.

Luego de encontrar el nodo inicial para el terreno en proceso, se realizará una comparación entre los valores de los nodos en dirección del nodo final, es decir, se estipularon 4 posibles rutas para tomar a través del terreno:

1. Desde un punto cercano al Noroeste hacia un punto cercano al Sureste.
2. Desde un punto cercano al Sureste hacia un punto cercano al Noroeste.
3. Desde un punto cercano al Noreste hacia un punto cercano al Suroeste.

4. Desde un punto cercano al Suroeste hacia un punto cercano al noreste.

Dependiendo del camino que tome podrá valorar los 2 nodos más próximos a ese punto (Si se dirige al Sureste, realiza la comparación entre su nodo más próximo al Sur y al Este). Eligiendo aquel que consuma una menor cantidad de combustible y cambiando su atributo de camino de un estado falso a uno verdadero.

Este proceso se repetirá hasta llegar a una de las excepciones: Que se encuentre en la misma fila o columna que la posición de inicio, o que ya haya llegado. En el primer caso entonces el movimiento del robot se limitará a simplemente moverse en sentido al punto de meta. En el segundo únicamente habrá terminado y cambiará su característica de camino a verdadera. Posteriormente imprimirá un mapa compuesto por las posiciones del terreno simbolizadas como un 0 para aquellas posiciones omitidas en la ruta final y 1 para aquellas incluidas, en este se podrá visualizar de manera detallada el camino seleccionado y que será el implementado por el robot. Seguidamente nos informará cuanta gasolina empleó en llevar a cabo el desplazamiento. Finalmente volverá al menú principal imprimiéndolo y solicitando una de las opciones.

Para imprimir el archivo de salida se hará una impresión de todos los terrenos que han sido procesados, luego se solicitará el nombre del terreno al usuario, el nombre introducido será buscado por medio de esta entrada en la lista enlazada principal. Sin embargo, existen restricciones para realizar esta tarea, no se podrá imprimir un archivo respecto a un terreno que no haya sido procesado o que no exista dentro de la lista enlazada principal.

Habiendo verificado que el terreno puede ser exportado a un XML, se buscará la información básica del terreno y junto con las librerías “etree” con sus propiedades ElementTree se hará la conversión de esta información de un formato de texto a una estructura XML adaptado y lista para la generación del documento final.

Solo la información principal acerca del recorrido será la introducida al archivo de salida. La información requerida será el nombre del terreno, las dimensiones

del terreno, las coordenadas del punto inicial y del punto final, la inversión en gasolina total del camino tomado y por ultimo las posiciones del recorrido que contaran con su coordenada y su inversión de combustible realizada por R2E2 para atravesarlo.

Esta información será ordenada por medio de las librerías ya mencionadas, asignándolas como sub elementos de un nodo padre el cual será el nombre del mismo terreno. Por último, en el proceso nos solicitara la ruta exacta en donde queremos que se guarde nuestro archivo XML de salida. El software nos indicara si ocurre algún tipo de error en este proceso, de lo contrario nos notificara imprimiendo un mensaje que el archivo XML se ha generado satisfactoriamente. Devolviéndonos al menú principal nuevamente.

“Generar graficas de terreno” utiliza un funcionamiento parecido a la impresión de archivos XML pues se mostrará un menú de los terrenos procesados y de seleccionarse un terreno fuera de las limitaciones (ser inexistente o no haber sido procesado) nos lo notificará solicitando nuevamente el nombre del terreno.

Las gráficas se realizaron generando primero un archivo de tipo dot el cual con ayuda de Graphviz se interpretará su contenido y devolverá una imagen con la terminación deseada (jpg o png).

Graphviz (Graph Visualization) es un conjunto de herramientas de software para el diseño de diagramas definido en el lenguaje descriptivo DOT.1 Fue desarrollado por “AT&T Labs” y liberado como software libre con licencia tipo Eclipse.

Graphviz consiste en un lenguaje de descripción de gráficos llamado DOT, un conjunto de herramientas y librerías que pueden generar o procesar archivos DOT.

Dot es una herramienta en línea de comandos para producir imágenes en capas de grafo dirigido en una variedad de formatos de salida (PostScript, pdf, svg, png, etc.).

Los archivos tipo DOT que esta extensión interpretara se generaron a través de la implementación de una cadena de texto la cual llevara

todo el contenido requerido, nodos y sus relaciones. Los nodos son inicializados con ayuda de una estructura cíclica de tipo “while” que hará el recorrido completo valuando si forman parte del recorrido, en caso de confirmación los nodos tendrán un borde de un color distinto al de aquellos que muestren una negativa ante la validación, el texto que mostrará cualquier tipo de nodo será el combustible.

Una vez llevados los nodos a la cadena de texto se asignarán las conexiones necesarias para que cada nodo este conectado a su norte, este, oeste y sur según sea el caso. Para ello se utilizaron dos ciclos “while” extra, uno para conexiones horizontales y el restante para conexiones en vertical, usando las conexiones existentes, recorriéndolas hasta llegar a alguno de los límites de nuestra pseudo matriz y al llegar guardan la coordenada contraria al movimiento regresando al nodo inicial para hacer tantos movimientos como este indicaba y volver a repetir la misma secuencia hasta tener las instrucciones necesarias para la generación del grafico representativo del terreno solicitado previamente por el usuario.

Se declararía con una última instrucción el tipo de imagen al cual se exportará toda la información, se indicará el archivo en el cual esta guardada la información y el nombre que se le asignara a la imagen generada.

Al finalizar el proceso de generación mostrara el mensaje de que el proceso se ha completado con satisfacción, de lo contrario mostrara un mensaje de error. Posteriormente nos devolverá al menú principal.

“Mostrar Datos del Estudiante” hará la impresión de como su nombre lo indica, los datos del estudiante autor de este software, imprimiendo su nombre y apellidos; la facultad a la que pertenece, Facultad de Ingeniería; su carne que lo identifica como parte de la institución; la carrera cursada: Ciencias y sistemas; y el semestre en el que se encuentra el matriculado. Posteriormente volveremos al menú principal, esperando la siguiente instrucción deseada por el usuario.

Las soluciones expuestas en el texto anterior fueron a través del método de prueba y error con la información y experiencia obtenida mediante investigación del uso de herramientas para Python y librerías o herramientas

externas.

El análisis y creación de nodos y estructuras de datos dinámicas se a utilizado por ser un recurso que haga relaciones de distintos tipos entre la información y poder acceder entre ellos de una manera práctica.

Conclusiones:

La solución presenta el algoritmo en el cual funcionan convencionalmente las matrices, lo que permite al usuario una familiarización adecuada para del uso del software.

El análisis y creación de nodos y estructuras de datos dinámicas se ha utilizado por ser un recurso que haga relaciones de distintos tipos entre la información y poder acceder entre ellos de una manera práctica.

Es importante modelar un problema antes de empezar a realizarlo, por medio de diagramas de flujo y clases para comprender, analizar y encontrar la solución de mejor manera.

Anexos:

Archivo de entrada solicitado

```
<terreno nombre="terreno2">
  <dimension>
    <n>2</n>
    <m>2</m>
  </dimension>
  <posicioninicio>
    <x>1</x>
    <y>1</y>
  </posicioninicio>
  <posicionfin>
    <x>2</x>
    <y>2</y>
  </posicionfin>
  <posicion x="1" y="1">2</posicion>
  <posicion x="1" y="2">1</posicion>
  <posicion x="2" y="1">3</posicion>
  <posicion x="2" y="2">5</posicion>
</terreno>
```

Figura 1. Archivos de Entrada

Fuente: Elaboración propia, 2021

Diagrama de Clases del software

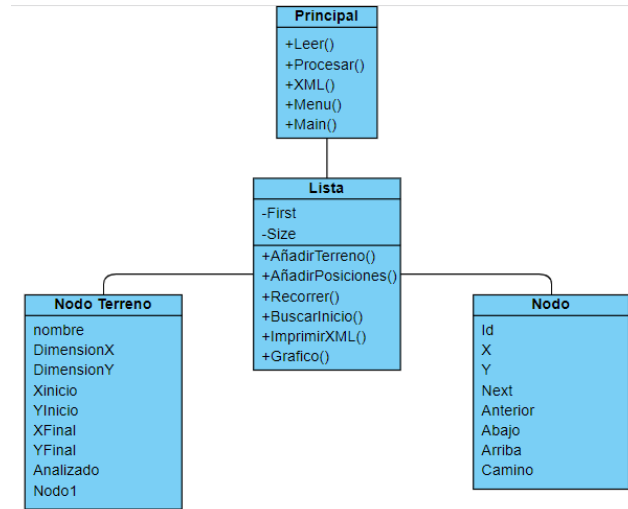


Figura II. Diagrama de Clases

Fuente: Elaboración propia, 2021