
Proyecto 2: Misiones de extracción de Civiles y Recursos en un enfrentamiento

202000544 – Roberto Carlos Gómez Donis

Resumen

Se realizó la creación de un programa para la creación de un mapa y realizar un camino para extraer recursos o civiles mediante el lenguaje Python implementando TDA's (Tipo de datos abstractos) para el manejo de listas enlazadas y doblemente enlazadas, POO (programación orientada a objetos) para lograr guardar información en su respectiva clase. La lectura de archivo xml para extraer información de entrada con la ayuda de la herramienta graphviz para obtener gráficamente los nodos que se crean al momento de ingresar el archivo.

Este software es de gran ayuda para al gobierno para una misión en caso de un conflicto, tanto por localizar el camino para extraer a los civiles o extraer recursos, esto se hará gráficamente.

Palabras clave

TDA, XML, NODO, LISTA ENLAZADA, MATRIZ DISPERSA.

Abstract

The creation of a program for the creation of a map and to make a path to extract resources or civilians through the Python language was carried out, implementing ADT's (Abstract Data Type) for the handling of linked and doubly linked lists, OOP (object-oriented programming). To be able to save information in its respective class. The reading of the xml file to extract input information with the help of the graphviz tool to graphically obtain the nodes that are created when entering the file.

This software is of great help to the government for a mission in the event of a conflict, both by locating the path to extract civilians or extract resources, this will be done graphically.

Keywords

TDA, XML, NODE, LINKED LIST, SPARSE MATRIX.

Introducción

La utilización de archivos XML en la actualidad es muy común entre los softwares, ya que se facilita el manejo de información en los archivos. Al momento de analizar se utiliza TDA para guardar la información de una manera óptima para procesar y lograr graficar los nodos en pdf.

Desarrollo del tema

El proyecto se divide en 4 partes, primero se realiza la lectura del archivo XML, esta es una de las partes más importante para que funcione el programa, ya que sin este no es funcional el programa. Este contendrá el nombre de la ciudad que se desea analizar, los robots que se tienen para realizar las misiones y coordenadas de los nodos que posteriormente se grafican dentro de una matriz dispersa, lo que nos lleva a la segunda parte que es la representación gráfica de una matriz dispersa usando graphviz. Este archivo podrá contener n cantidad ciudades y robots. Una vez leído el archivo se hace el uso de TDA, en este caso es una lista simple enlazada, para guardar datos generales de la ciudad y una lista simple enlazada para guardar los robots que tenga disponible el archivo, además y más importante se crea una clase Matriz Dispersa, donde se guardan todos los tipos de casillas que puede contener la ciudad que son 'E' para puntos de entrada, '*' para casillas intransitables, 'C' para unidades civiles y 'R' para casillas donde se encuentran recursos

a. Nodo:

En informática y en telecomunicaciones, de forma muy general, un nodo es un punto de intersección, conexión o unión de varios elementos que confluyen en el mismo lugar.

b. Lista Simplemente enlazada:

Es una lista enlazada de nodos, donde cada nodo tiene un único campo de enlace. Una variable de referencia contiene una referencia al primer nodo.

c. Matriz dispersa:

Es una lista enlazada que a diferencia de la simple esta tiene dos enlaces, donde hace referencia al anterior y al siguiente.

d. Graphviz:

Graphviz (Graph Visualization) es un conjunto de herramientas de software para el diseño de diagramas definido en el lenguaje.

Como primera función es leer el archivo XML y cargarlo a las listas para poder manipular los datos escritos en cada archivo.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuracion>
  <listaCiudades>
    <ciudad>
      <nombre filas="10" columnas="10">CentralCity</nombre>
      <fila numero="1">**** C***</fila>
      <fila numero="2">** * *****</fila>
      <fila numero="3">** * ** R</fila>
      <fila numero="4">E **</fila>
      <fila numero="5">***** **</fila>
      <fila numero="6">R **</fila>
      <fila numero="7">** **** C</fila>
      <fila numero="8">** *** **</fila>
      <fila numero="9">** *** **</fila>
      <fila numero="10">***E*** **</fila>
      <unidadMilitar fila="2" columna="5">20</unidadMilitar>
      <unidadMilitar fila="6" columna="6">35</unidadMilitar>
    </ciudad>
  </listaCiudades>
</configuracion>
```

Figura 1. Título o descripción breve de la figura.
Fuente: Elaboración Propia 2022

```
<robots>
  <robot>
    <nombre tipo="ChapinFighter" capacidad="150">Peñador1</nombre>
    <nombre tipo="ChapinRescue">Rescatador1</nombre>
    <nombre tipo="ChapinFighter" capacidad="50">Peñador2</nombre>
    <nombre tipo="ChapinFighter" capacidad="70">Peñador3</nombre>
    <nombre tipo="ChapinRescue">Rescatador2</nombre>
    <nombre tipo="ChapinRescue">Rescatador3</nombre>
  </robot>
</robots>
```

Figura 2. Título o descripción breve de la figura.

Fuente: Elaboración Propia 2022

Luego de obtener la información se crean los nodos, dichos nodos crean un archivo con extensión “.txt”, aunque también se puede crear en diferentes extensiones, pero la que se utilizó para el software fue “.txt”.

```
digraph TD
  node[shape=box fontname="Arial" fillcolor="white" style=filled width=1.98 height=1.98 ]
  edge[style = invis]
  root[label = "capa: None", group=1]
  label = "MATRIZ DISPERSA"
  fontname="Arial Black"
  fontsize="15pt"

  F1[label="F1", fillcolor="plum", group=1];
  F2[label="F2", fillcolor="plum", group=1];
  F3[label="F3", fillcolor="plum", group=1];
  F4[label="F4", fillcolor="plum", group=1];
  F5[label="F5", fillcolor="plum", group=1];
  F6[label="F6", fillcolor="plum", group=1];
  F7[label="F7", fillcolor="plum", group=1];
  F8[label="F8", fillcolor="plum", group=1];
  F9[label="F9", fillcolor="plum", group=1];
  F10[label="F10", fillcolor="plum", group=1];
  F11[label="F11", fillcolor="plum", group=1];
  F12[label="F12", fillcolor="plum", group=1];
  F13[label="F13", fillcolor="plum", group=1];
  F14[label="F14", fillcolor="plum", group=1];
  F15[label="F15", fillcolor="plum", group=1];
  F16[label="F16", fillcolor="plum", group=1];
  F17[label="F17", fillcolor="plum", group=1];
```

Figura 2. Título o descripción breve de la figura.

Fuente: Elaboración Propia 2022

```
F17->F16;
F17->F18;
F18->F17;
F18->F19;
F19->F18;
F19->F20;
F20->F19;
C1[label="C1", fillcolor="powderblue", group=2];
C2[label="C2", fillcolor="powderblue", group=3];
C3[label="C3", fillcolor="powderblue", group=4];
C4[label="C4", fillcolor="powderblue", group=5];
C5[label="C5", fillcolor="powderblue", group=6];
C6[label="C6", fillcolor="powderblue", group=7];
C7[label="C7", fillcolor="powderblue", group=8];
C8[label="C8", fillcolor="powderblue", group=9];
C9[label="C9", fillcolor="powderblue", group=10];
C10[label="C10", fillcolor="powderblue", group=11];
C11[label="C11", fillcolor="powderblue", group=12];
C12[label="C12", fillcolor="powderblue", group=13];
C13[label="C13", fillcolor="powderblue", group=14];
C14[label="C14", fillcolor="powderblue", group=15];
C15[label="C15", fillcolor="powderblue", group=16];
```

Figura 2. Título o descripción breve de la figura.

Fuente: Elaboración Propia 2022

Luego que se crea el archivo “.txt” se convierte en extensión “.pdf” para graficar los nodos, todo con ayuda de la herramienta de graphviz, sin embargo, hay que tener en cuenta que para que se grafique bien, los nodos deben de ir enlazados de forma correcta, ya que se puede dar la ocasión que se enlacen de forma incorrecta y quede de manera esperada.

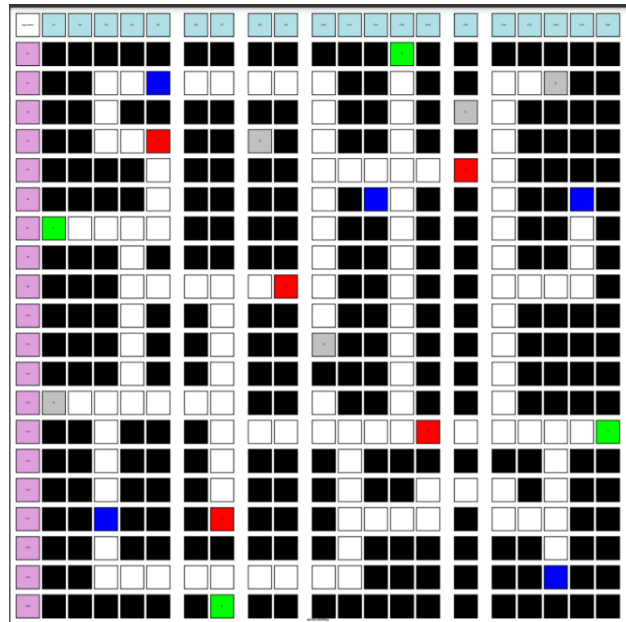


Figura 3. Título o descripción breve de la figura.

Fuente: Elaboración Propia 2022

Conclusiones

- Los Tipos de Datos Abstractos(TDA) tienen una funcionalidad vital a la hora de trabajar con memoria dinámica ya que optimiza el uso de la memoria.
- La herramienta Graphviz es eficiente a la hora de graficar grafos y TDA's
- Los Archivos xml son vitales a la hora de transferir datos.

Referencias bibliográficas

- Teh, P. S., Zhang, N., Teoh, A. B. J., & Chen, K. (2016). TDAS: a touch dynamics based multifactor authentication solution for mobile devices. International Journal of Pervasive Computing and Communications.
- Gansner, E. R. (2009). Drawing graphs with Graphviz. Technical Report, Technical Report

Diagrama de clase

