

AUTOMATIZACIÓN DE RESCATE CIVIL Y EXTRACCIÓN DE RECURSOS

201809935 – Saúl Alfredo Barbero Contreras

Resumen

La aplicación tiene como objetivo ayudar a la empresa “Chapín Warriors, S.A.” en el rescate de civiles y la extracción de recursos en ciudades que se encuentran en conflictos bélicos.

Para poder llevar a cabo el análisis se debe de ingresar un archivo con terminación .xml que contiene una ciudad, las rutas transitables, las unidades civiles y las unidades con recursos. Además de poseer el conocimiento de las unidades militares en la ciudad.

Se utilizó un algoritmo adaptado del algoritmo de *Dijkstra* y el algoritmo *A-Star*. Estos algoritmos buscan la ruta más corta de un punto a otro punto en la matriz Ortogonal.

La solución permite consultar los reportes generados de cada misión realizada por los robots de la empresa.

Palabras clave

XML, Dijkstra, A-Star, Matriz Ortogonal, Nodo.

Abstract

The application is intended to help the company "Chapín Warriors, S.A." in the rescue of civilians and the extraction of resources in cities that are in warlike conflicts.

To carry out the analysis, an .xml file containing a city, passable routes, civilian units, and units with resources. In addition to possessing the knowledge of military units in the city.

An algorithm adapted from Dijkstra's algorithm and the A-Star algorithm was used. An algorithm adapted from Dijkstra's algorithm and the A-Star algorithm was used.

The solution allows you to consult the reports generated from each mission carried out by the company's robots.

Keywords

XML, Dijkstra, A-Star, Orthogonal Matrix, Node

Introducción

El análisis de datos es un aspecto muy importante, para hallar la solución a cualquier problemática. En este ensayo vamos a resolver la problemática que plantea la empresa “*Chapín Warriors, S.A.*” Halla una ruta que sea transitable para alguna de sus robots. Estos robots pueden ser de rescate o de poder militar. La ruta es importante para que el centro de comando pueda generar una estrategia durante el combate.

Para hallar esta ruta se creó un algoritmo basado en el algoritmo A y el algoritmo Dijkstra, estos hallan el camino más corto y eficiente de un punto a otro punto.

Desarrollo del tema

La finalidad de la aplicación es solucionar la problemática expuesta: hallar el camino más corto. Suena algo sencillo de realizar, sin embargo, se debe de llevar a cabo una serie de pasos o algoritmo para llegar a la solución requerida.

Estructura de la entrada.

El archivo de entrada debe de ser un archivo xml. Este es un lenguaje de marcado, similar a lo que es HTML, donde su información esta alojada entre etiquetas.

```
<ciudad>
  <nombre filas="15" columnas="20">CiudadGotica</nombre>
  <fila numero="1">*****</fila>
  <fila numero="2">**** *          *</fila>
  <fila numero="3">**** *****E*****C*****</fila>
  <fila numero="4">**** ***** ***** **R*</fila>
  <fila numero="5">E          *</fila>
  <fila numero="6">**** * * * * * * *</fila>
  <fila numero="7">**** * * * * * **R*</fila>
  <fila numero="8">*          *</fila>
  <fila numero="9">**** * * * * * * *</fila>
  <fila numero="10">**** * * * * * * *</fila>
  <fila numero="11">****          *</fila>
  <fila numero="12">**** * * * * * * *</fila>
  <fila numero="13">**** * * E * * **R *****</fila>
  <fila numero="14">**** *****          C*</fila>
  <fila numero="15">**** *****C***** *****</fila>
  <unidadMilitar fila="5" columna="14">20</unidadMilitar>
  <unidadMilitar fila="5" columna="19">20</unidadMilitar>
  <unidadMilitar fila="5" columna="20">20</unidadMilitar>
  <unidadMilitar fila="8" columna="16">20</unidadMilitar>
  <unidadMilitar fila="11" columna="7">20</unidadMilitar>
  <unidadMilitar fila="11" columna="15">20</unidadMilitar>
  <unidadMilitar fila="14" columna="10">20</unidadMilitar>
</ciudad>
```

Figura 1. Ejemplo de un archivo de entrada.

Fuente: elaboración propia.

Entrada y salida de datos.

La entrada del archivo xml se realiza ingresando la ruta del archivo con su extensión en la consola al momento de desplegar la aplicación.

La salida de información se genera en un PDF. Este PDF contiene una imagen generada mediante graphviz y textos descriptivos de las misiones realizadas.

Un archivo PDF es un formato de almacenamiento para documentos digitales independientes de plataformas de software o hardware. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).¹ Es multiplataforma, es decir, puede ser utilizado en los principales sistemas operativos (GNU/Linux, MacOS, Unix, Windows), sin que se modifique el aspecto ni la estructura del documento original.

Graphviz es un conjunto de herramientas de software para el diseño de diagramas definido en el lenguaje descriptivo DOT. Fue desarrollado por AT&T Labs y liberado como software libre.²

Dot es una herramienta de línea de comando para generar imágenes tipo grafos en una gran variedad de formatos

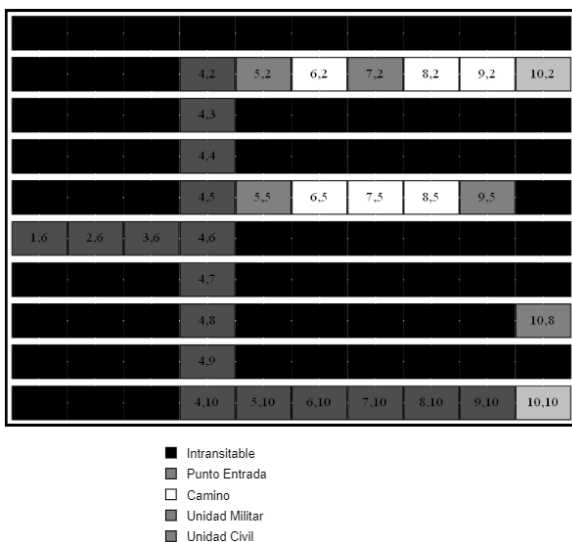


Figura II. Ejemplo de salida.

Fuente: elaboración propia.

Solución.

¿Cómo encontrar una ruta para llegar a nuestro objetivo? A partir de esta pregunta se procedió a realizar la solución.

Solo podemos movernos de manera ortogonal, es decir en dirección norte para ir hacia arriba, sur para ir hacia abajo, oeste para ir a la izquierda y este para ir a la derecha.

Cada nodo o posición contiene parámetros específicos según el tipo que sea declarado en la entrada del programa. Zonas transitables o no transitables, unidades militares con una capacidad de combate y zonas con civiles o recursos que se desean recuperar.

Al procesar el archivo de entrada se crea por defecto una matriz ortogonal con cada una de las ciudades que se desea realizar una misión. Estas ya cargadas con cada nodo en su posición específica.

El algoritmo utilizado es una implementación de dos algoritmos con lógicas diferentes pero que cumplen una misma finalidad.

Un algoritmo que iniciara una búsqueda de los nodos de arriba, derecha, abajo, izquierda. Y verificar si es un nodo transitable y un algoritmo generador de un grafo, donde se almacena las posibles rutas antes visitadas y así determinar una ruta hacia nuestro objetivo. Esto dará como resultado un algoritmo de fuerza bruta.

Conclusiones

Un algoritmo de fuerza bruta encontrara una solución sin importar que halla recorrido caminos erróneos.

La complejidad de los algoritmos de fuerza bruta es exponencial esto hace que no sean los algoritmos más eficientes.

Referencias bibliográficas

1. Adobe.com. 2022. *¿Qué es un PDF? Formato de documento portátil | Adobe Acrobat DC.* [online] Available at:
<<https://www.adobe.com/la/acrobat/about-adobe-pdf.html>> [Accessed 6 April 2022].
2. Graphviz. 2022. *Graphviz.* [online] Available at:
<<http://www.graphviz.org/>> [Accessed 6 April 2022].