SISTEMA AUTÓNOMO DE "CHAPIN WARRIORS S.A."

201908355 – Danny Hugo Bryan Tejaxún Pichiyá

Resumen

A medida que crece la empresa y el avance de la tecnología continua, se requiere que el proceso de rescate de unidades civiles y extracción de recursos sea eficiente sin poner en riesgo a más personas en la ciudad del conflicto. Por lo que se necesita de la implementación de algoritmos de búsqueda de rutas de escape y estrategias de combate.

El programa carga la información mediante archivos de entrada, además de contar con una funcionalidad de búsqueda mediante nombre de la ciudad o robot.
Al buscar rutas de escape de unidades civiles se busca llevarlos a salvo fuera de la zona de guerra evitando cualquier tipo de enfrentamiento con unidades militares ubicadas en distintos puntos estratégicos de la ciudad. También se busca extraer recursos valiosos de la ciudad tratando de eliminar unidades militares si el robot de extracción tiene las condiciones suficientes para combatir.

Es posible afirmar que el sistema cumple de manera eficiente con su función, y que demuestra que la misión pueda completarse o no.

Palabras clave
Robot, Unidad Civil, Unidad Militar, Ciudad, Camino

Abstract

The company and technology are changing so it is need that the process of civil units rescue and resources extraction be successful without endanger to more people in warzone. Then it is need implement of optimization algorithms of pathfinding for scape and combat strategies.

The program load data through input also it has search function through city name or drone name.

To search path to scape of civil units it is need take the people to safety outside of warzone and avoid any type of clash with military units located in different places tactically in the city. Also it is need extract resources valuable of the city trying dispose of military units if the drone has the enough conditions for fight.

Is possible say that the system is doing its function and prove that the mission can do it or can't do it.

Keywords
Drone, Civil Unit, Military Unit, City, Path

Introducción

La utilización de robots en la extracción de personas civiles o recursos valiosos de zonas de guerra se ha convertido en la solución más viable para evitar el mayor número de decesos posibles.

Chapin Warriors S.A. ha implementado un novedoso sistema que encuentra rutas para enviar un robot a buscar personas atrapadas en alguna ciudad con zonas de guerra, además de que dichas rutas también pueden ser halladas para poder ir a una zona con recursos valiosos y poder extraerlos.

El sistema de generación de instrucciones se manipula con un menú en consola en el que se pueden realizar ciertas acciones adicionales.

Desarrollo del tema

El sistema se basa en listas enlazadas conteniendo nodos y objetos según sea el caso. El concepto de lista enlazada se basa en el almacenamiento de nodos cuyos atributos permiten contener información y en ciertos casos otros nodos para poder tener acceso a la información y poder manipularla a conveniencia.

El sistema será manejado mediante consola y la implementación de un menú, de tal modo que sea fácil de usar y sea agradable al usuario. El programa seguirá en funcionamiento de manera continua hasta que el usuario desee salir seleccionando la respectiva opción del menú.

El sistema tiene una funcionalidad para generar un archivo que contiene una gráfica de la ciudad con los puntos importantes en dónde se encuentran puntos de entrada, unidades civiles, unidades militares o puntos de entrada, así como los caminos disponibles, la ciudad se selecciona de las ingresadas mediante un archivo de entrada con formato XML con toda la información relevante para el caso.

a) Carga de Datos

Los datos se cargan en el sistema ingresando la ubicación del archivo XML, conteniendo la información de las ciudades con sus respectivos nombres y unidades militares y civiles, puntos de entrada y recursos. Además, el archivo contiene información de los robots que realizarán las misiones.

b) Ejecución de Misiones

Los dos tipos de misiones son de Rescate y de Extracción de Recursos. Las misiones se realizan seleccionando primero el robot y luego la ciudad. Si la misión es de rescate la puede realizar únicamente un robot de tipo ChapinRescue, si no hay robots de este tipo no pueden realizarse estas misiones. Si la misión es de extracción la puede realizar únicamente un robot de tipo ChapinFighter, si no hay robots de este tipo no pueden realizarse estas misiones.

c) Graficación

Es posible seleccionar una ciuad y automáticamente la función de graficación generará un documento en formato pdf en el que se encuentra la gráfica de la ciudad con sus puntos críticos, además de la descripción del tipo de cada posición en la ciudad.

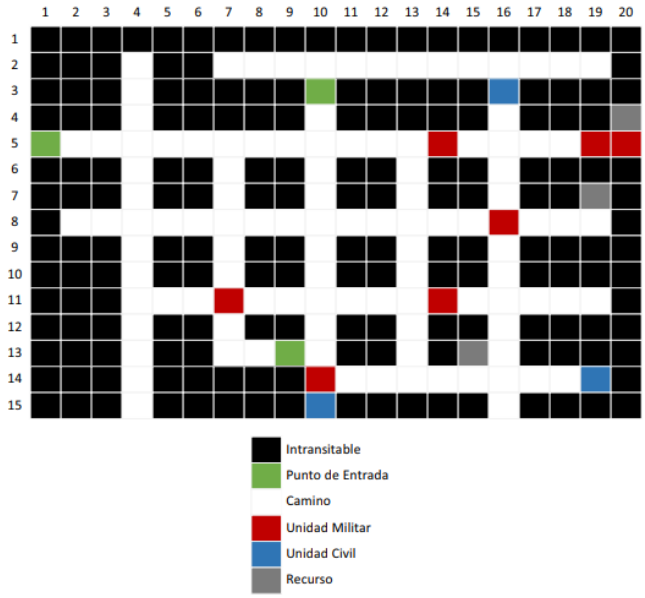

Figura 1. Ejemplo de gráfica de Ciudad generado por el sistema.

Fuente: Elaboración propia.

d) Algoritmo de Búsqueda de Camino

Al seleccionar todos los parámetros necesarios para realizar una misión (robot, ciudad y destino) el algoritmo buscará caminos hacia el destino partiendo de un punto de entrada de la ciudad. El sistema priorizará el punto más cercano al punto de destino. Se toman en cuenta únicamente las primeras 250 caminos posibles (si fuesen más). Se consideran todos los escenarios posibles de forma simultánea, es decir, que se evalúa cuando vaya hacia arriba, abajo, izquierda o derecha al mismo tiempo, todos los caminos alternos coexisten simultáneamente, y únicamente se guardan los caminos que lleguen hacia el punto de destino. El algoritmo funciona mediante llamadas recursivas del método. En el caso de que la misión es de extracción, se evalúa si la posición siguiente está ocupada por una unidad militar en el caso de que sí se le reta la capacidad militar a la capacidad de combate del robot y para volver a hacer la llamada recursiva se evalúa de que su capacidad de combate actual sea mayor que cero.

e) Generación de Reporte de Misión

Al finalizar la misión se generará un reporte en formato pdf en el que se muestra gráficamente la ciudad y el camino que tomó el robot hasta llegar a su destino, el reporte se genera para cualquier tipo de misión, en dado caso que la misión sea imposible de completar se hará la aclaración en el reporte. Se detallan aspectos de la misión como qué robot realizó la misión, qué tipo de misión fue (rescate o extracción de recursos), las coordenadas del destino, la capacidad de combate inicial del robot (si la misión es de extracción de recursos) y la capacidad de combate final del robot (si la misión es de extracción de recursos).

Figura 2. Ejemplo de Misión de Rescate generado por el sistema.

Fuente: Elaboración propia.

Lógica del Sistema

Toda la información es almacenada en listas enlazadas, y en el caso de las listas enlazadas que contienen unidades militares son atributos del constructor correspondiente, ya que una ciudad contiene uno o más unidades civiles, de igual forma con las listas que contienen cada posición de la ciudad, ya que una ciudad contiene 1 o más posiciones.

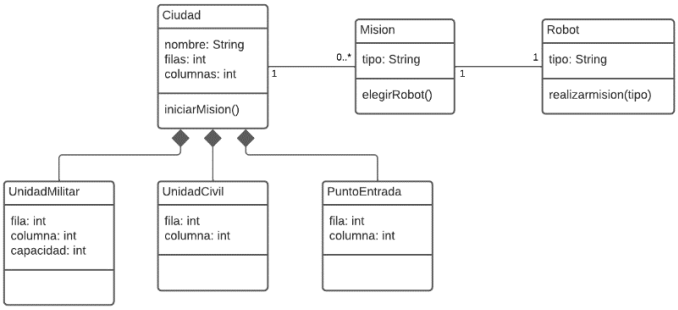

Figura 3. Diagrama de Clases.

Fuente: Elaboración propia.

En el diagrama se puede observar de manera más completa la organización lógica de la codificación del sistema.

Las distintas listas tienen diferentes funciones para insertar, buscar, etc. información, de modo que el manejo sea más sencillo.

Algunos de los métodos más importantes de las listas son:

1. **insert**: Permite crear un nodo en la posición primera de la lista, y para las posteriores posiciones, se insertará los nuevos nodos en el atributo siguiente del último nodo.
2. **search**: Permite localizar un nodo en específico mediante un parámetro de búsqueda (nombre), mediante el uso de bucles y comparaciones hasta coincidir con el parámetro recibido.
3. **get**: En la lista enlazada de patrones, permite obtener información específica de un nodo como por ejemplo el nombre de la ciudad, las posiciones de la ciudad, etc, mediante el envío de índices (coordenadas o posiciones).

Conclusiones

Luego de comprender las funciones del sistema y de las clases implementadas se puede afirmar que el sistema si cumple de forma eficaz con la búsqueda de caminos.

Para el manejo de listas enlazadas es posible incluir ciertos atributos a las clases y a sus nodos, de modo que el acceso a su contenido no sea muy complicado y a la vez se convierte en una estructura de almacenamiento de información bastante útil y fácil de manipular.

El uso de la herramienta Graphviz permite la visualización de la ruta que el algoritmo encontró en caso de que la misión haya sido completada, de modo que es posible ver una ruta eficiente para completar la función. En caso de que la función no haya sido completada únicamente se verá la ciudad con los detalles de la misión.

Referencias bibliográficas

(s.f.). Obtenido de CodinGame: https://www.codingame.com/playgrounds/7656/los-caminos-mas-cortos-con-el-algoritmo-de-dijkstra/el-algoritmo-de-dijkstra

Developer. (s.f.). Obtenido de https://developer.apple.com/library/archive/documentation/General/Conceptual/GameplayKit_Guide/Pathfinding.html

Guzmán, D. A. (s.f.). Obtenido de http://opac.pucv.cl/pucv_txt/txt-5000/UCE5372_01.pdf

López, B. S. (s.f.). Obtenido de Ingenieria Industrial: https://www.ingenieriaindustrialonline.com/investigacion-de-operaciones/algoritmo-de-dijkstra/