|  |
| --- |
| **DOCUMENTACIÓN PROYECTO 2** |
| **201700900 – Gueslim Jhoans Fernandez Rivera** |

**Resumen**

Se desarrolló un programa para controlar equipos automatizados para rescatar civiles y extraer recursos de las ciudades que se encuentran inmersas en conflictos bélicos el fin de realizar las misiones de rescate y extracción para Chapín Warriors, S. A. Para poder completar las misiones de rescate o extracción de recursos de ciudades en conflicto, se tiene dos tipos de drones. ChapinRescue: Unidad robótica capaz de ingresar a la ciudad en uno de los puntos de entrada, seguir un camino seguro (celdas transitables) hasta la “unidad civil”. Y ChapinFighter: Unidad robótica capaz de ingresar a la ciudad en uno de los puntos de entrada, seguir un camino hasta un “recurso” y extraerlo.

**Palabras clave**

Rescate, misión, pathfinding, Python, TDA, POO

***Abstract***

*A program was developed to control automated equipment to rescue civilians and extract resources from cities that are immersed in armed conflicts in order to carry out rescue and extraction missions for Chapin Warriors, S. A. In order to complete rescue missions or extraction of resources from cities in conflict, there are two types of drones. ChapinRescue: Robotic unit capable of entering the city at one of the entry points, following a safe path (passable cells) to the "civilian unit". And ChapinFighter: Robotic unit capable of entering the city at one of the entry points, following a path to a "resource" and extracting it.*

***Keywords***

*Tile, cost, cost, optimization, Python, TDA, POO*

**Carga de XML**

Todos las configuraciones de ciudades y robots vienen almacenados en un archivo de formato XML, por lo que el primer paso para que el programa funcione es cargar y almacenar el archivo.

Para esto se utilizó el módulo xml.etree.ElementTree, con el que se iterando para raíz del piso en que se fueron tomando los datos.

**Lista Enlazada**

Lo más esencial y el núcleo del funcionamiento del programa es la creación de un tipo de dato abstracto para el almacenamiento y el procesamiento de la información obtenido del archivo XML. Para ello se eligió crear una Lista Enlazada Simple con POO. Esta lista enlazada consta de una clase Nodo, y una clase LinkedList como se aprecia en la figura 1.

Diagrama

Descripción generada automáticamente

*Figura 1.* Diagrama de clases general.

Fuente: elaboración propia, o citar al autor, año y página.

**Ciudad**

La clase Ciudad consta de todas las etiquetas del archivo XML como atributos. Esta clase almacena el nombre de ciudad, la malla de la ciudad, el número de filas y columnas. La lista de patrones correspondientes a in piso se van guardando en una lista enlazada, siendo cada nodo un objeto Patrón.

Esta clase es muy importante debido a que almacena toda la información de cada ciudad, así como grafica la información almacenada para mostrar al usuario.

**Militar**

La clase Militar almacena la posición en (x, y) de la unidad militar, así como su capacidad de combate.

**Robot**

La clase Robot tiene como atributos el nombre del dron, el tipo y la capacidad de combate.

**Uso**

Al correr el programa aparecerá el menú de las distintas opciones enumeradas, por lo cual debemos elegir una opción presionando uno de los números.

Texto

Descripción generada automáticamente

1. Cargar XML

Al elegir la primera opción, nos pedirá la ruta del archivo XML, luego de introducirla se almacenarán los datos en el sistema y nos regresará al menú principal.

Texto

Descripción generada automáticamente

1. Mostrar Patrón.

Al elegir la segunda opción, si aún no se han elegido el piso y patrón origen, se desplegarán todos los pisos y sus patrones disponibles en el sistema.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Por lo que se deberá introducir el nombre del piso, o el número de piso según es listado, se desplegarán los patrones disponibles de ese piso, y el sistema le pedirá que introduzca del código de patrón o el numero según es listado.

Texto

Descripción generada automáticamente

Luego de seleccionar el Piso y patrón iniciales, regresará al menú principal. Al seleccionar la opción 2 nuevamente luego de elegir el piso y patrón inicial se verá la opción elegida, y el sistema le pedirá que confirme si desea cambiarlos presionando S, dando nuevamente las opciones iniciales.

1. Seleccionar nuevo Patrón.

En esta opción se desplegará el piso actual elegido anteriormente, y el sistema le pedirá que ingrese el código del patrón destino o el numero según listado.

Luego de esto se desplegará el procedimiento paso a paso del procedimiento para transformar del patrón actual al nuevo. Y por último de desplegará una imagen del grafico del patrón original y el nuevo.

1. Ordenar pisos Cargados

Al seleccionar esta opción la lista de pisos y patrones en el sistema de ordenaran alfabéticamente según el nombre del piso y el código del patrón. Desplegando todos los datos ordenados.

**Referencias bibliográficas**

Charles Severance (2015). *Python para Informaticos.* CreateSpace Independent Publishing Platform.

**ANEXOS EN EL FOLDER**