
SISTEMA DE GESTIÓN Y PROCESAMIENTO DE MENSAJES PARA DRONES

202200174 – Andres Alejandro Quezada Cabrera

Resumen

El Ministerio de Defensa de Guatemala y la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala han establecido un acuerdo para desarrollar una tecnología de envío de mensajes encriptados. Este sistema emplea drones que emiten luces LED a alturas específicas, representando letras del alfabeto.

El receptor decodifica el mensaje basándose en las alturas y nombres de los drones. Cada día, se genera una tabla de correspondencia entre alturas, letras y drones. Para transmitir un mensaje, se envían instrucciones de encendido de luces a los drones. Se busca desarrollar un software que maneje este sistema de drones eficientemente, considerando hasta 200 drones y alturas de 1 a 100 metros. El objetivo es decodificar el mensaje en el menor tiempo posible. Este sistema ofrece flexibilidad en la representación de letras, permitiendo varias alturas y drones para una misma letra. El receptor reconstruye el mensaje utilizando la tabla proporcionada por el Ministerio de Defensa.

Palabras clave

TDA (Tipos de Dato Abstracto), Matriz, Nodo, Lista Simplemente Enlazada, Tuplas.

Abstract

The Ministry of Defense of Guatemala has partnered with the Faculty of Engineering at the University of San Carlos of Guatemala to develop secure communication technology. This technology aims to enable the transmission of encrypted messages that can only be interpreted by authorized individuals or institutions. To achieve this, they have devised a system with two essential components: a message emitter and a message receiver. The proposal involves the use of "n" drones capable of ascending to different heights and emitting high-intensity LED light representing letters of the alphabet.

The receiver deciphers the message based on the height and the emitting drone. Each day, a key correspondence table is created. The goal is to develop software that efficiently manages this system, translating instructions into the final message in the shortest time possible while adhering to the established rules for this innovative form of communication. This project focuses on using Python and data structures to achieve this functionality.

Keywords

DTA (Abstract Data Types), Matrix, Node, Simply Linked List, Tuples.

Introducción

En Guatemala, el Ministerio de Defensa ha colaborado con la Facultad de Ingeniería de la Universidad de San Carlos para crear un revolucionario sistema de mensajería encriptada. Este proyecto, basado en Python, implica el uso de drones capaces de emitir luces LED a alturas específicas, representando letras del alfabeto. El receptor descifra el mensaje mediante la interpretación de alturas y drones. Cada día se genera una tabla que asocia alturas, letras y drones. El objetivo es desarrollar un eficiente software para manejar hasta 200 drones y alturas de 1 a 100 metros, decodificando el mensaje en el menor tiempo posible.

Desarrollo del tema

- Implementación de POO y TDA
- Lectura de archivos de entrada (Formato XML)
- Creación de archivos de salida (Formato XML)
- Operaciones con TDA
- Generar un grafo de una matriz seleccionada

A. Modulo - Interfaz.py

Importaciones:

Importa los módulos necesarios, como tkinter para la interfaz gráfica y otros para operaciones con archivos y XML.

Clase Interfaz:

Inicializa la interfaz gráfica y define métodos para cargar, gestionar drones, gestionar sistemas de drones, gestionar mensajes, y generar archivos XML.

Métodos principales:

cargar_frame: Crea y muestra la ventana de la interfaz gráfica.

salir: Cierra la ventana y finaliza el programa.

cargar_archivo_xml: Permite cargar un archivo XML y mostrar un mensaje de confirmación.

gestion_drones: Muestra una ventana para la gestión de drones.

agregar_dron: Agrega un dron a la lista de drones y actualiza la interfaz gráfica.

ventana_gestion_sistemas: Muestra una ventana para la gestión de sistemas de drones.

gestion_sistemas_drones: Genera gráficos de sistemas de drones utilizando Graphviz.

gestion_mensajes: Muestra una ventana para la gestión de mensajes y sus instrucciones.

on_select_tree: Maneja la selección de un mensaje en la interfaz y muestra sus instrucciones.

graficar_mensajes: Genera gráficos de mensajes y sus instrucciones utilizando Graphviz.

generar_archivo: Genera un nuevo archivo XML con los datos procesados.

calcular_datos: Calcula datos relacionados con los mensajes y sistemas de drones.

mover_dron: Simula los movimientos de un dron a cierta altura.

B. Modulo – ListasDobles.py

Clases y Estructuras de Datos:

Nodo: Representa un nodo en la lista, con referencias al siguiente y anterior nodo, y contiene información relacionada con drones, alturas, letras e instrucciones.

Movimientos: Representa un movimiento de un dron en un tiempo específico, con información sobre el dron, la altura, la letra y la instrucción.

ListaDoble: Es una lista doble que contiene nodos de tipo `Nodo_regular` que representan drones y sus movimientos. Tiene funciones para agregar nodos, actualizar movimientos, organizar la lista y graficar la información.

Operaciones en la Lista Doble:

Agregar nodos con información sobre drones y movimientos en diferentes alturas.

Actualizar instrucciones de un dron en una altura específica.

Graficar la información de los drones y sus movimientos en una tabla HTML.

Otras Clases:

`ListaAlturas`, `ListaSistemaDrones`, `ListaContenido`, `ListaInstruccion`, `ListaMensaje`, `ListaMovimiento`, `ListaMsgProcess`: Representan listas con diferentes tipos de información relacionada con alturas, sistemas de drones, contenido, instrucciones, mensajes, movimientos y mensajes procesados.

Funcionalidades Adicionales:

Obtener información sobre alturas, tiempo, contenido, movimientos, mensajes procesados, etc.
Generar un archivo XML con información sobre los mensajes procesados.

Interacción con Datos:

Las listas contienen información relacionada con drones, alturas, contenido, instrucciones y mensajes. Se pueden agregar, buscar, organizar y visualizar datos en estas listas.

Representa un nodo de la lista doble para almacenar información sobre drones, alturas, letras e instrucciones.

Tiene referencias al siguiente (`next`) y anterior (`back`) nodo en la lista.

Clase Nodo:

Representa un nodo de una lista simple y se utiliza para almacenar cualquier tipo de dato (`tipo_dato`) en la lista.

Clase Mensaje:

Representa un mensaje y almacena su nombre, el sistema asociado y una lista de instrucciones.

Clase MensajeProcesado:

Representa un mensaje procesado y almacena su nombre, mensaje, sistema asociado, tiempo y una lista de movimientos.

Clase Altura:

Representa una altura específica y almacena su valor, dron asociado y número de contenido.

Clase Instruccion:

Representa una instrucción asociada a un dron y almacena el dron y la instrucción en sí.

Clase Contenido:

Representa el contenido asociado a un dron y almacena el dron y una lista de alturas.

Clase SistemaDrones:

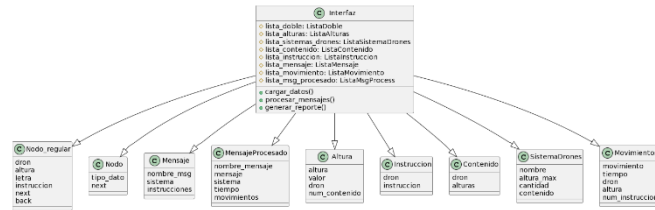
Representa un sistema de drones y almacena su nombre, altura máxima, cantidad de drones y una lista de contenidos.

Clase Movimientos:

Representa un movimiento de un dron en un tiempo específico y almacena el movimiento, tiempo, dron, altura y número de instrucción.

C. Modulo – Nodos.py

Clase `Nodo_regular`:



su interacción en un sistema determinado, facilitando la manipulación de información, generación de mensajes procesados y análisis de movimientos en alturas específicas.

Conclusiones

El código proporcionado define una serie de clases que representan elementos clave para el manejo y procesamiento de información en un sistema de drones. Se han establecido estructuras como nodos, mensajes, alturas, instrucciones, contenidos y movimientos, todos esenciales para llevar a cabo la gestión de mensajes, movimientos y estados de drones en un entorno dado.

La clase `NodoRegular` se utiliza para crear una lista doblemente enlazada, permitiendo almacenar información detallada sobre los drones, alturas, letras e instrucciones. Por otro lado, la clase `Nodo` se emplea para una lista simple y puede contener diversos tipos de datos a través de su atributo `tipo_datos`.

Se destaca la representación de mensajes y mensajes procesados mediante las clases `Mensaje` y `MensajeProcesado`, respectivamente. Estas clases contienen información relevante sobre los mensajes, sus sistemas asociados, tiempos y movimientos correspondientes.

Además, se han definido estructuras para representar alturas, instrucciones, contenidos y sistemas de drones, permitiendo organizar y acceder a la información de manera efectiva.

En conclusión, el código proporciona una estructura sólida y modular para abordar la gestión de drones y

Referencias bibliográficas

"Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein

"Data Structures and Algorithms in Python" by Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser