
PROYECTO No. 2

202200252 – Néstor Enrique Villatoro Avendaño

Resumen

El ministerio de la Defensa de Guatemala ha creado un acuerdo de colaboración con la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. El objetivo de este acuerdo es crear una nueva tecnología que permita el envío de mensajes encriptados de tal forma que no puedan ser interceptados y descifrados por personal o instituciones no autorizadas. Para lograr este fin, se ha creado un sistema con 2 componentes. Un componente emisor del mensaje y un componente receptor del mensaje.

Guatemala. The objective of this agreement is to create a new technology that allows the sending of encrypted messages in such a way that they cannot be intercepted and decrypted by unauthorized personnel or institutions. To achieve this end, a system with 2 components has been created. A message sending component and a message receiving component.

Palabras clave

Máximo cinco palabras que servirán para identificar el estudio realizado.

Keywords

Traducción al idioma inglés de las palabras clave.

Abstract

The Ministry of Defense of Guatemala has created a collaboration agreement with the Faculty of Engineering of the University of San Carlos of

Introducción

La Facultad de Ingeniería propone un diseño que funcionará con “n” drones, los cuales podrán subir una cantidad de metros y emitir una luz led de alta emisión de tal forma que dependiendo del dron que emita la luz y la altura a la que la emita, representará una letra del alfabeto, de esta forma, el componente receptor podrá medir alturas y determinar qué dron emitió el haz de luz, decodificar cada letra y finalmente obtener el mensaje que se desea transmitir.

Desarrollo del tema

El ministerio de la Defensa de Guatemala ha creado un acuerdo de colaboración con la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. El objetivo de este acuerdo es crear una nueva tecnología que permita el envío de mensajes encriptados de tal forma que no puedan ser interceptados y descifrados por personal o instituciones no autorizadas. Para lograr este fin, se ha creado un sistema con 2 componentes. Un componente emisor del mensaje y un componente receptor del mensaje. La Facultad de Ingeniería propone un diseño que funcionará con “n” drones, los cuales podrán subir una cantidad de metros y emitir una luz led de alta emisión de tal forma que dependiendo del dron que emita la luz y la altura a la que la emita, representará una letra del alfabeto, de esta forma, el componente receptor podrá medir alturas y determinar qué dron emitió el haz de luz, decodificar cada letra y finalmente obtener el mensaje que se desea transmitir. Cada día, el ministerio de defensa creará una tabla como la siguiente:

7	K	N	Q
6	J	M	P

5	I	L	O
4	F	G	H
3	D	P	E
2	I	C	C
1	A	2	B
Altura	DronX	DronY	DronZ

Cada dron es capaz de realizar las siguientes acciones:

1. Subir 1 metro
2. Bajar 1 metro
3. Esperar
4. Emitir luz de alta energía

El sistema de drones debe respetar las siguientes reglas:

- a. Solamente un dron puede “Emitir luz de alta energía” en un tiempo “t” dado.
- b. Un dron puede emitir la misma letra a distintas alturas, así será más difícil hackear el sistema de encriptación producido por los drones.
- c. Un dron demora 1 segundo en subir o bajar un metro y demora 1 segundo en encender y apagar la luz de alta energía.

Para crear un mensaje a transmitir, simplemente se transmite al sistema de drones una secuencia de instrucciones que determina el orden en que deben encender sus luces cada uno de los drones y la altura a la que lo deben hacer. Así, por ejemplo, si el mensaje es “Hello World” se podría enviar la secuencia de instrucciones que se muestra en la tabla 1 al sistema de drones:

DronX,2 - DronX a 2 metros (representa la I)
DronY,3 - DronY a 3 metros (representa la P)
DronZ,2 - DronZ a 2 metros (representa la C)
DronY,1 - DronY a 1 metros (representa la 2)

Debido a que un sistema de drones puede tener una letra en varias alturas y/o en diferentes drones, entonces, un mismo mensaje podría ser enviado con distintas instrucciones. El sistema receptor, será capaz de detectar las alturas y los nombres de los drones, de esta forma, podrá buscar en el sistema de drones proporcionado por el Ministerio de Defensa, la letra correspondiente y reconstruir el mensaje. Usted ha sido contratado por la Facultad de Ingeniería de USAC para desarrollar un Software capaz de manejar el sistema de drones que genera el mensaje, y en base a las instrucciones recibidas, generar el mensaje en el menor tiempo posible. El sistema deberá ser capaz de manejar hasta 200 drones y alturas de 1 a 100 metros.

Instrucciones detalladas que se deben enviar al sistema de drones para que emita el mensaje encriptado (optimizando el tiempo para generar el mensaje)

Tiempo	DronX	DronY	DronZ
1	subir	subir	subir
2	subir	subir	subir
3	Emitir luz	subir	esperar
4	esperar	Emitir luz	esperar
5	esperar	bajar	Emitir luz
6	esperar	bajar	esperar
7	esperar	Emitir luz	esperar

En este ejemplo, el tiempo óptimo para emitir el mensaje haciendo que el sistema de drones encienda sus luces en las alturas determinadas por el set de instrucciones es de 7 segundos. El mensaje enviado es el siguiente: IPC2

Se podrá utilizar un archivo en formato XML para crear una configuración inicial que facilite la evaluación de la funcionalidad del software desarrollado.

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <listaDrones>
    <dron>DronX</dron>
    <dron>DronY</dron>
    <dron>DronZ</dron>
  </listaDrones>
  <listaSistemasDrones>
    <sistemaDrones nombre="SD1">
      <alturaMaxima>7</alturaMaxima>
      <cantidadDrones>3</cantidadDrones>
      <contenido>
        <dron>DronX</dron>
        <alturas>
          <altura valor="1">A</altura>
          <altura valor="2">I</altura>
          <altura valor="3">D</altura>
          <altura valor="4">F</altura>
          <altura valor="5">I</altura>
          <altura valor="6">J</altura>
          <altura valor="7">K</altura>
        </alturas>
      </contenido>
      <contenido>
        <dron>DronY</dron>
        <alturas>
          <altura valor="1">2</altura>
          <altura valor="2">C</altura>
          <altura valor="3">P</altura>
          <altura valor="4">G</altura>
          <altura valor="5">L</altura>
          <altura valor="6">M</altura>
          <altura valor="7">N</altura>
        </alturas>
      </contenido>
      <contenido>
        <dron>DronZ</dron>
        <alturas>
          <altura valor="1">B</altura>
          <altura valor="2">C</altura>
          <altura valor="3">E</altura>
          <altura valor="4">H</altura>
        </alturas>
      </contenido>
    </sistemaDrones>
  </listaSistemasDrones>
</config>
```

```
<altura valor="5">0</altura>
<altura valor="6">P</altura>
<altura valor="7">Q</altura>
</alturas>
</contenido>
</sistemaDrones>
</listaSistemasDrones>
<listaMensajes>
  <Mensaje nombre="msg">
    <sistemaDrones>SD1</sistemaDrones>
    <instrucciones>
      <instruccion
dron="DronX">2</instruccion>
      <instruccion
dron="DronY">3</instruccion>
      <instruccion
dron="DronZ">2</instruccion>
      <instruccion
dron="DronY">1</instruccion>
    </instrucciones>
  </Mensaje>
</listaMensajes>
</config>
```

Para extraer y ordenar la información se utilizó la raíz del documento y se guardaron los datos en listas simples con el dato y el índice. Posterior a eso se procedió a guardar los datos de forma ordenada en listas dobles y/o listas de listas, de tal modo que quedara una lista doble adentro de un objeto con un nombre el cual sería del dron, esto para las alturas de los drones. Luego esas listas se guardarían en una lista de listas con un nombre una altura máxima y una cantidad de drones y posteriormente esa lista de listas se guardaría en una lista de listas para almacenar todos los sistemas de drones. Ejemplo gráfico en anexos.

Para la lista de mensajes se hizo un procedimiento similar, se guardaron los drones y sus alturas dentro de una lista simple, luego se guardó esa lista en un objeto con un nombre y un sistema de drones a utilizar. Para poder tener varias instrucciones, se

guardaron en una lista de listas. Ejemplo gráfico en anexos.

- a. Inicialización – Para que el sistema pueda inicializarse sin ninguna información previa.
- b. Cargar un archivo XML de entrada
- c. Generar un archivo XML de salida
- d. Gestión de drones
 - a. Ver listado de drones ordenado alfabéticamente
 - b. Agregar un nuevo dron (debe ser un nombre único en el software)
- e. Gestión de sistemas de drones
 - a. Ver gráficamente listado de sistemas de drones (utilizando Graphiz)
- f. Gestión de Mensajes
 - a. Ver listado de mensajes y sus instrucciones (ordenado alfabéticamente por nombre del mensaje)
 - b. Ver instrucciones para enviar un mensaje
 - i. Seleccionar un mensaje
 - ii. Mostrar el nombre sistema de drones a utilizar, el mensaje que se enviará y el tiempo óptimo para que el sistema de drones muestre el mensaje
 - iii. Ver gráficamente (utilizando Graphiz) el listado de instrucciones que se debe enviar al sistema de drones para que genere el mensaje en un tiempo óptimo.
 - g. Ayuda – Mostrar la información del estudiante y un link hacia la documentación del proyecto

Se generó un archivo en formato XML que muestre el detalle de instrucciones para enviar los mensajes configurados en los distintos sistemas de drones y el tiempo óptimo necesario para enviar dichos mensajes. Ejemplo en anexos.

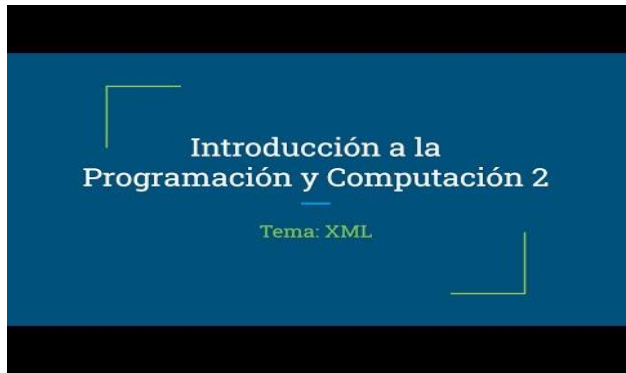
Conclusiones

En conclusión, Python es un lenguaje muy efectivo para modelar, documentar e implementar una solución al problema que se plantea utilizando las herramientas de desarrollo como Graphviz, tkinter, lectura xml, POO, TDAs, etc.

En este proyecto se cumplieron los siguientes objetivos: Implementar una solución utilizando el lenguaje de programación Python. Utilizar estructuras de programación secuenciales, cíclicas y condicionales. Generar reportes con la herramienta Graphviz. Manipular archivos XML. Utilizar los conceptos de TDA y aplicarlos a memoria dinámica. Utilizar estructuras de programación propias. Utilizar el paradigma de programación orientada a objetos.

Referencias bibliográficas

[Daniel Alfaro]. (2023, August 15). IPC2 Sección C - Clase XML [Video]. YouTube. [IPC2 Sección C - Clase XML](#)



[Daniel Alfaro]. (2023, August 27). IPC2 Sección C - Clase Graphviz [Video]. YouTube. <https://youtu.be/tLXmvsDxio?si=K0TpG43igS6ryiPm>

Welcome to. (s/f). Python.org. Recuperado el 7 de septiembre de 2023, de <https://www.python.org/>

Graphviz. (s/f). Graphviz. Recuperado el 7 de septiembre de 2023, de <https://graphviz.org/>

Extensions, L. M. A. (s/f). Visual Studio Code - code editing. Redefined. Visualstudio.com. Recuperado el 7 de septiembre de 2023, de <https://code.visualstudio.com/>

Anexos

Ejemplo gráfico lista sistema de drones

lista Sistema Drones											
nombre, altura, cantidad				nombre, altura, cantidad				nombre, altura, cantidad			
DronX	DronY	DronZ		DronX	DronY	DronZ		DronX	DronY	DronZ	
1-A	1-A	1-A		1-A	1-A	1-A		1-A	1-A	1-A	
2-1	2-1	2-1		2-1	2-1	2-1		2-1	2-1	2-1	
3-D	3-D	3-D		3-D	3-D	3-D		3-D	3-D	3-D	
4-F	4-F	4-F		4-F	4-F	4-F		4-F	4-F	4-F	
5-1	5-1	5-1		5-1	5-1	5-1		5-1	5-1	5-1	
6-1	6-1	6-1		6-1	6-1	6-1		6-1	6-1	6-1	
7-E	7-E	7-E		7-E	7-E	7-E		7-E	7-E	7-E	

Ejemplo gráfico lista mensajes

lista mensajes		
nombre, sistema	nombre, sistema	nombre, sistema
DronX - 2	DronX - 2	DronX - 2
DronY - 3	DronY - 3	DronY - 3
DronZ - 2	DronZ - 2	DronZ - 2
DronY - 1	DronY - 1	DronY - 1

Ejemplo salida

```
<?xml version="1.0"?>
<respuesta>
  <listaMensajes>
    <mensaje nombre="nombreMensaje">
      <sistemaDrones>[valorAlfanumerico]</sistemaDrones>
      <tiempoOptimo>[valorNumerico]</tiempoOptimo>
      <instrucciones>
        <tiempo valor="valorNumerico">
          <acciones>
            <dron nombre="valorAlfanumerico"> [valorAccionDron] </dron>
            ...
          </acciones>
        </tiempo>
        ...
      </instrucciones>
    </mensaje>
    ...
  </listaMensajes>
</respuesta>
```

Diagrama de clases

