Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2021.

PROYECTO 1 – INTRODUCCION A LA PROGRAMACION Y COMPUTO 2

201700375 – Bryan Steve Montepeque Santos

Resumen

De parte de la Agencia Guatemalteca de Investigación Espacial (AGIE) se ha enviado a un robot de exploración llamado R2E2 a un planeta para explorar nuevos terrenos, debido a las condiciones del planeta R2E2 siempre debe de elegir el camino que menor cantidad de combustible utilice para poder volver a la base, R2E2 solo puede moverse ortogonalmente, es decir solo hacia el Norte, Sur, Este y Oeste debido a esto se han dividido los terrenos en una cuadricula y mediante avanzados procesos matemáticos se determinó la cantidad de combustible que tomaría recorrer cada uno de esos terrenos, esto es un proyecto en conjunto con la Universidad del Valle y su satélite el Quetzal-01, lo que la AGIE necesita de la Universidad de San Carlos es que diseñe el programa para el movimiento de R2E2 y para que la ruta que tome siempre sea la ruta más eficiente con el combustible.

Palabras clave

1. Lista Simple
2. Nodo
3. XML
4. Grafo
5. Matriz Ortogonal

Abstract

The Guatemalan Agency of Spatial Investigation (GASI) has sent an exploration robot called R2E2 to a planet in order to explore new land, due to the conditions of the planet R2E2 must always go through the route that uses the least amount of fuel possible in order to be able to get back to the base, however R2E2 can only move orthogonally, this is, only in four directions: North, South, East and West because of this the land has been divided into a grid and using advanced mathematical processes it has been determined how much fuel would be spent on each of the areas, this is joint project between Universidad del Valle and their satellite Quetzal-01, so what the GASI requires the Universidad de San Carlos is for it to design the program for R2E2's movement so that it always takes the most fuel efficient route possible.

Keywords

1. *Linked List*
2. *Node*
3. *XML*
4. *Graph*
5. *Orthogonal Linked List*

Introducción

Debido a lo caro que es desarrollar robots espaciales de alta capacidad el R2E2 cuenta con memoria limitada, entonces se requiere que el sistema para el manejo de las matrices sea altamente eficiente y que no se desperdicie ningún byte de memoria, para esto se ha llegado a la conclusión de que se deben utilizar Tipos de Datos Abstractos para implementar las listas y matrices que necesite, el realizar este proyecto representa un gran avance no solo para el país sino para la humanidad.

Desarrollo del tema

Para poder desarrollar este programa el proceso fue dividido en varios pasos:

1. Realizar el Menú:
   Este paso es bastante sencillo, solamente consiste en desplegar un conjunto de opciones y pedir que se ingrese una opción de todas las opciones que vé en la pantalla siendo estas:
   - Cargar el Archivo
   - Procesar el Archivo
   - Escribir Archivo Salida
   - Mostrar Datos del Estudiante
   - Generar Gráfica
   - Salir

   Para poder reconocer qué opción ingresó el usuario usaremos un input, y al leer su respuesta es comparada con un if else de varios casos para saber cuál de los métodos ejecutar, y este a su vez está encerrado por un try catch en caso de que el usuario decida ingresar un carácter invalido o lo haga por error

2. Cargar el Archivo XML:
   Para esto se le pide al usuario que ingrese la ruta absoluta del archivo incluyendo su extensión, si el archivo elegido no es un XML entonces muestra error y regresa al menú, pero si sí es un XML entonces utiliza la librería Element Tree para leer todas las etiquetas del archivo XML y las va identificando, luego de esto al identificar un Terreno almacena su nombre así como sus dimensiones, y posiciones iniciales y finales de la ruta que se desea seguir así como los datos de cada cuadro del terreno, los datos del terreno son guardados en una Matriz Ortogonal para que en el proceso posterior sea más fácil trabajarlos, así mismo facilita el ingresar los datos del XML a la matriz, luego toma todos estos datos del terreno, las dimensiones, la posición inicial de X y Y, la posición final de X y Y y los datos del terreno en un nodo de una Lista Enlazada. Y va imprimiendo los valores de cada etiqueta que reconoce en consola, al terminar se muestra un mensaje y le pide que ingrese cualquier valor para volver al menú principal.

3. Procesar el Archivo:
   Aquí es en donde se calcula la ruta más eficiente para que R2E2 recorra, ya que se tienen diferentes terrenos para recorrer se le pide al usuario que ingrese cuál de los diferentes terrenos desea procesar, una vez ha elegido uno se procede a buscar ese terreno en la Lista de Terrenos, y se toman los datos de ese terreno así como la matriz que contiene la cuadricula del Terreno elegido y aquí debido a que era más sencillo se genera todo el código que irá en el archivo de salida, de esta forma se evita volver a hacer este proceso de nuevo, entonces, se toman todos los datos del terreno y se crea un “Nodo Actual” que representa la posición de R2E2 e inicialmente se coloca en la posición inicial de X y Y que indica el archivo XML cargado para este terreno para

esto fue utilizada la función "MostrarNodoMatriz" la cual recibe como parámetros una Fila y una Columna y devuelve el nodo ubicado en esa posición de la Matriz Ortogonal esto facilitó mucho el trabajo a la hora de programar, entonces se recorre toda la matriz y mediante el uso de varias condiciones para replicar el Algoritmo de Djikstra el cual muestra cómo obtener la ruta más corta para llegar de un nodo a otro y al mismo tiempo ya que estamos generando la ruta en este paso se va generando simultanea mente el código XML de la Ruta Utilizada para llegar de un nodo a otro, debido a esto se ha simplificado el proceso de Escribir el Archivo de Ruta, todo este proceso se hace mostrando en consola varias notificaciones del proceso actual que se está realizando, al final se almacena en otra matriz el nuevo mapa que solo contiene los nodos de la ruta que fue utilizada por R2E2, esto es para poder mostrar en consola un mapa de la ruta que tomó el robot, para esto solo se recorre la nueva matriz que solo contiene la ruta y se utilizan "0" para representar por donde no pasó R2E2 y "1" para representar cada cuadro por donde sí pasó, una vez esto está graficado se muestra el nombre del terreno que se recorrió y la cantidad de combustible empleada, luego de esto le pide ingresar cualquier valor para volver al menú principal.

4. Escribir Archivo Salida:
   Se debe escribir un archivo XML utilizando los datos de la ruta que utilizó R2E2 para poder moverse a través del terreno para esto lo primero que se hace es pedirle al usuario la ruta de donde desea que se genere el archivo, una vez ha ingresado la ruta solo toma el código que fue generado en el paso de "Procesar Archivo" y almacenado en una variable Global y lo escribe en un archivo llamado "Proyecto1AS.xml" para poder asegurar que este se almacene en la ruta ingresada con el usuario se utiliza un join para unir la ruta del usuario con un "/" seguido del nombre del archivo junto con su extensión, una vez este proceso ha sido realizado se genera el archivo, muestra una notificación en consola de que el archivo ya fue generado y por ultimo pide ingresar cualquier valor para volver al menú principal.

5. Mostrar los Datos del Estudiante:
   Bien, solo se usan varios String para mostrar los datos del desarrollador de la aplicación, en este caso se mostrarán:
   - Nombre Completo
   - Carnet
   - Nombre Completo del Curso
   - Carrera del Estudiante
   - A qué semestre de dicha carrera pertenece el curso

6. Graficar las Matrices:
   En esta parte debemos de generar una gráfica en GraphViz de la Matriz que nos solicite el usuario, para esto solo debemos de desplegar todos las matrices almacenadas en nuestra Lista de Terrenos y pedirle al usuario que ingrese el número de la matriz que quiere ver graficada y con un if else comparar el valor que ingrese el usuario con el id del terreno en la Lista Enlazada, si este id es igual al de cualquier terreno en la lista circular entonces ya procede a graficarla si no es igual al id de ninguna matriz en la lista circular entonces debe mostrarle al usuario que ingreso mal el dato o bien que no se encontró ninguna matriz con ese número, para esto debemos utilizar un try catch, que si en dado caso el valor no es ninguno de los valores que se muestran tire una excepción, y vuelva a mostrar el menú y a pedir una opción hasta que el usuario ingrese una opción válida, después de esto, se procederá a graficar la

matriz seleccionada para esto se utilizará un String que vaya concatenando el código de GraphViz (El cuál no verá el usuario, esto ya va escrito en el código del programa) y al mismo tiempo que este string se va concatenando también se va recorriendo la Matriz seleccionada por el usuario para poder obtener los datos correctos en cada fila y columna, así como también ir haciendo las conexiones correctas entre cada uno de los nodos de la matriz. Después de concatenar todo el String, así como en el inciso anterior debemos de escribirlo todo a un archivo llamado "TerrenoGraficado.dot" y utilizaremos la librería os para usar su método system() para enviar comandos a la consola de Windows y de esta forma generar la imagen de la gráfica que deseamos ver y que posteriormente automáticamente abra la gráfica de la matriz generada.

7. Salida:
   Cierra el programa usando sys.exit() que cierra el programa de Python y termina todos los procesos abiertos en el momento.

Conclusiones

El programa fue dividido en varios "Mini Programas" o módulos para poder simplificarnos la tarea de programarlo y también para poder repartirnos el trabajo entre varias personas si fuera necesario hacerlo así, esta dedicación y atención al detalle se ve reflejada a lo largo de todo el programa, empezando por el método del menú donde fue necesario usar una condición que verificara si una opción era válida y después usar un while que repitiera el ciclo try catch ya que sin el while el bloque try solo se ejecuta una vez, entonces fue necesario encerrarlo todo en un while y escribirle dentro del if else a cada opción válida que cambiara el valor de la variable de False a True para que el código del while dejara de repetirse, este nivel de calidad lo hemos puesto en todas las partes del programa así que creemos que utilizarlo será una experiencia agradable para nuestros usuarios.

Referencias bibliográficas

1. GeeksforGeeks. (2018, August 31). *Linked List Data Structure*. https://www.geeksforgeeks.org/data-structures/linked-list/

2. *Definición de nodo — Definicion.de*. (2008, November 8). Definición.de. https://definicion.de/nodo/#:%7E:text=La%20programaci%C3%B3n%20inform%C3%A1tica%20considera%20que,de%20referencia%20para%20otro%20nodo.

3. *XML introduction - XML: Extensible Markup Language | MDN*. (2021, February 19). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction

4. O. (2020, March 10). *Qué son los grafos*. GraphEverywhere. https://www.grapheverywhere.com/que-son-los-grafos/

5. GeeksforGeeks. (2021, June 14). *Orthogonal Linked List*. https://www.geeksforgeeks.org/orthogonal-linked-list/

Diagrama de Clases

Bryan Steve Montepeque Santos | August 29, 2021