EXPLORADOR

201801295 – José Rodrigo Garcia Godínez

Resumen

La Agencia Guatemalteca de Investigación Espacial (AGIE) diseño un robot de exploración. Este explorara nuevos terrenos y necesita la ayuda de poder calcular la ruta más factible atreves de los terrenos escaneados con anterioridad. Estos terrenos se darán con la ruta más factible y con el consumo que el robot tiene que hacer para poder llegar a su destino.

El diseño se basó en el comportamiento del plano cartesiano o el movimiento de datos en una matriz. Esto se dio ya que el robot solo puede moverse en dirección principal de los 4 puntos cardinales (Norte, Sur, Este, Oeste).

Todos los datos serán mandados por medio del satélite Quetzal01. Este satélite será el encargado de poder dar visibilidad al robot desde las alturas y así poder completar todos los datos para llegar al destino prometido.

Palabras clave

Simplificar para dar resultados rápidos.

.

Abstract

The Guatemalan Agency for Space Research (AGIE) designed an exploration robot. It will explore new terrain and needs the help of being able to calculate the most feasible route through the previously scanned terrain. These lands will be found with the most feasible route and with the consumption that the robot has to do in order to reach its destination.

The design was based on the behavior of the Cartesian plane or the movement of data in a matrix. This occurred since the robot can only move in the main direction of the 4 cardinal points (North, South, East, West).

All data will be sent through the Quetzal01 satellite. This satellite will be in charge of being able to give visibility to the robot from above and thus be able to complete all the data to reach the promised destination.

Keywords

Simplify to give fast results.

Introducción

El proyecto fue elaborado en Python por ayuda de módulos implementados por el mismo compilador para poder simplificar procesos y ayudar al programador en su elaboración. Se utilizó los paradigmas POO, imperativo, declarativo y secuencial para llevar a cabo todas las funciones que el proyecto ofrece al usuario final. Se utilizó una interfaz amigable y entendible para que el usuario final comprenda como acceder a los datos de cada opción.

Desarrollo del tema

El proyecto consistió en un gestor de terrenos con el proceso de rutas deseadas con el aproximado de combustible que un robot tomara en avanzar hasta su destino. El diseño principal se hizo por medio de procesos en los puntos cardinales principales, con ello se almaceno en matrices los datos para llevar un mejor control en los movimientos y el combustible que se usara en dicha posición (x,y).

Menu

El menú principal consta de 6 opciones donde se puede navegar para obtener los resultados.

Cargar Archivo

Carga los datos recibidos por medio de un archivo XML (con el propósito de tener un control de los tipos de datos y poder trabajar por medio de una programación orientada a objetos) enviados por el satélite y cargados por una persona o actualizados en el sistema por el mismo robot. Este mostrara cuando los datos estén cargados con éxito.

Procesar archivo

Procesa el terreno seleccionado para poder escribir los archivos de salida y dar a entender al robot los cuadrantes que tiene que moverse para llegar a su posición final. Esto despliega todos los terrenos cargados para poder seleccionar un terreno específicamente y cargados para su extracción por medio de un archivo XML.

Escribir archivo de salida

Escribe un archivo XML con los datos procesados en la opción 2 donde se mostrará los datos del terreno procesado, el combustible que se gastara en recorrer desde el inicio hasta el final y las posiciones que este tiene que pasar para poder ahorrar combustible ya que cada posición del terreno tiene un estimado de combustible que se puede gastar para salir.

Mostrar datos del estudiante

Mostrara los datos del estudiante de la Facultad de ingeniería de la Universidad de San Carlos de Guatemala que realizo el programa.

Generar gráfico

Genera un gráfico por medio de la herramienta de graphiz donde se muestra por medio de una matriz los valores del combustible denominados por el satélite que el robot tiene que usar para poder entrar y salir de dicho cuadrante.

Paradigma

Se utilizo el paradigma orientado a objetos para guardar y manejar todos los datos en el sistema y así poder tener un acceso rápido a cada dato cargado por medio el archivo .xml. Otro paradigma usado con mayoridad es el paradigma estructural, ya que se declaró cada funcionen su momento haciendo que el programa cambie de función o de rumbo en la ejecución del código.

Otro paradigma es el declarativo, ya que se usaron operaciones matemáticas para realizar operaciones en

las matrices. Esto se llevó a cabo por medio de los módulos math y numpy, esto se llevó para agilizar procesos de llenado de matrices, cálculos comparativos entre datos, entre otros.

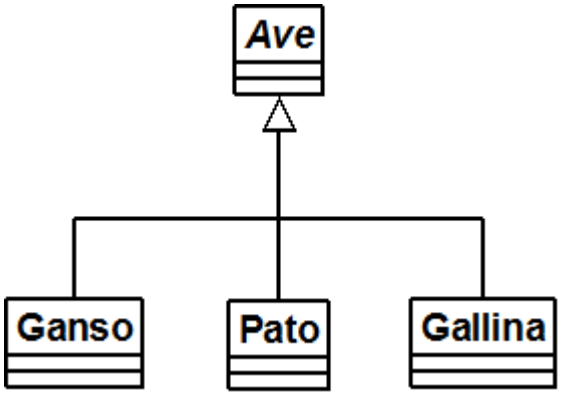


Ilustración 1Ejemplo de POO

Sistema

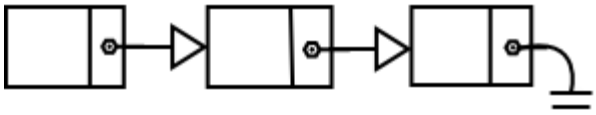
Objetos creados en base a la programación orientada a objetos.

Funcionalidad del TDA

Un Tipo de dato abstracto (en adelante TDA) es un conjunto de datos u objetos al cual se le asocian operaciones. El TDA provee de una interfaz con la cual es posible realizar las operaciones permitidas, abstrayéndose de la manera en cómo estén implementadas dichas operaciones.

El paradigma de orientación a objetos permite el encapsulamiento de los datos y las operaciones mediante la definición de clases e interfaces, lo cual permite ocultar la manera en cómo ha sido implementado el TDA y solo permite el acceso a los datos a través de las operaciones provistas por la interfaz.

Estructura de un nodo



Lista enlazada

Ilustración 2Ejemplo de una Lista Simple

Graphiz

Graphviz consiste en un lenguaje de descripción de gráficos llamado DOT, un conjunto de herramientas y librerías que pueden generar o procesar archivos DOT.

Esta herramienta fue usada para poder llegar a visualizar las matrices cargadas por medio del archivo .xml. Esta cumple la función principal de hacer un gráfico que lleva al usuario final a ver las llaves y poder realizar cambio.

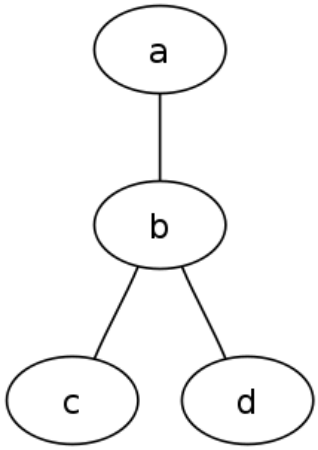


Ilustración 3Ejemplo de graphiz

Conclusiones

Se resolvió el problema con mayor facilidad usando módulos proporcionado por Python para tener un mejor control y acceso a los datos cargados. Se utilizó una lista genérica para no tener problemas con datos que no sean semejantes o hereditarios por el uso de la programación orientada a objetos.

Referencias bibliográficas

Briega, L. R. E. (2015, 19 julio). Expresiones Regulares
con Python. Matemáticas, análisis de datos y python.
https://relopezbriega.github.io/blog/2015/07/19/expres
iones-regulares-con-python/

Listas doblemente enlazada circular. (s. f.). CódigoFacilito. Recuperado 8 de marzo de 2021, de https://codigofacilito.com/videos/listas-doblementeenlazada-circular

3.9. Tipo listas — Materiales del entrenamiento de programación en Python - Nivel básico. (s. f.). COVANTEC. Recuperado 8 de marzo de 2021, de https://entrenamiento-pythonbasico.readthedocs.io/es/latest/leccion3/tipo_listas.htm
l

User Guide — graphviz 0.16 documentation. (s. f.). Graphviz. Recuperado 8 de marzo de 2021, de https://graphviz.readthedocs.io/en/stable/manual.htm
l

Welcome to. (s. f.). Python.org. Recuperado 8 de marzo
de 2021, de https://www.python.org/doc/