MANUAL TECNICO

GIAN CARLOS FIGUEROA

JORGE ALEJANDRO AGUIRRE

LUZ ENITH GUERRERO

DOCENTE

PROYECTO

UNIVERSIDAD DE CALDAS

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

ANALISIS Y DISEÑO DE ALGORITMOS

MANIZALES CALDAS

MAYO DEL 2017

MANUAL TECNICO

|  |  |  |
| --- | --- | --- |
| Clase | Descripción | Métodos |
| AlgoritmosRuta | Contiene el algoritmo de Floyd Warshall, que permite calcular la ruta mínima entre los nodos de un grafo. | * LlenarPesos: Permite llenar la matriz de pesos para calcular el mínimo camino. * floydWarshall: Contiene las operaciones propias de la filosofía del algoritmo de Floyd warshall para calcular caminos minimos. * obtenerCamino: Apartir de la matriz calculada de Floyd Warshall, se obtiene el camino de nodos. |
| AreaItems | Contiene una lista de los elementos que puede seleccionar el usuario para construir la ciudad e interrupciones | * CargarComponente: Llena una lista con los componentes que puede seleccionar el usuario en la interfaz para construir la ciudad. |
| Item | Elemento con atributos como área y tipo de imagen que puede seleccionar el usuario para construir la ciudad. |  |
| Componente | Representa cada elemento con los que se puede construir la ciudad por lo tanto contiene un nombre, una ruta de la imagen, una área y un id si se considera nodo. |  |
| Interrupción | Indica que un componente es del tipo interrupción, almacena el componente que está por debajo de la interrupción para recuperarlo después de eliminar la interrupción |  |
| Ciudad | Contiene una matriz de los componentes que tiene la ciudad, cantidad de nodos, y lista de interrupciones. | * EsCalle, esCarretera, esVia, esViaCortada, esCruce: Verifica si el tipo de componente pertenece a una calle, carretera o cruce. * MarcarNodosAdyacentes: Cuando el usuario crea una interrupción en la ciudad, se ejecuta este método para marcar a los componentes adyacentes a la interrupción como nodo. * EliminarNodosAdyacentes: Al eliminarse una interrupción se deben eliminar los nodos adyacentes marcados a la misma. * actualizarCiudad: Cuando se carga la ciudad desde un archivo de texto, se debe reconstruir sus propiedades. * modificarNodos: modifica los id de los nodos de menor a mayor. (Para un mejor debug). * marcarNodo: Asigna a un componente un id de nodo. |
| Carro | Contiene la imagen que representa el vehiculo, el área del carro dentro del mapa y posee una lista de aristas que es el camino que el carro tomara. |  |
| CarroMovimiento | Contiene la ciudad, el grafo la ubicación origen y los nodos de destinos | * Run: Permite cambiar las coordenadas de movimiento del vehículo según el camino que tiene asignado. * PuedoPasar: Verifica si es posible pasar por un cruce (Semaforos inteligentes). * ReconstruirUbicacion y reconstruirDestinos: Permite obtener los id de nodos de los destinos que fueron cambiados cuando el usuario creo alguna interrupción en la ciudad. |
| Arista | Contiene las coordenadas de inicio y fin de la arista y una lista de componente que conforman la arista como también la velocidad a la que se puede transitar, longitud y tráfico de la arista. |  |
| RutaCorta | Hereda de AlgoritmoRuta, Permite calcular la ruta más corta sobrescribiendo el método llenarPesos con la longitud de cada uno de las aristas. | LlenarPesos: A partir del grafo se obtiene la longitud de cada arista. |
| RutaVeloz | Hereda de AlgoritmoRuta, Permite calcular la ruta más corta sobrescribiendo el método llenarPesos con la velocidad de cada una de las aristas. | LlenarPesos: A partir del grafo se obtiene la velocidad en la que se puede transitar en cada arista. |
| RutaTrafico | Hereda de AlgoritmoRuta, Permite calcular la ruta más corta sobrescribiendo el método llenarPesos con la cantidad de tráfico que hay en cada una de las aristas. | LlenarPesos: A partir del grafo se obtiene la velocidad en la que se puede transitar en cada arista. |