Instituto Tecnológico de Costa Rica

Unidad de Computación

II Proyecto Programado

Estructuras de Datos

Steven Peraza Porras

Eliomar Rodríguez Arguedas

Alberth Salas Calero

Sede San Carlos

Noviembre 2016

Tabla de Contenido

[1. Introducción 2](#_Toc467528460)

[2. Análisis del Problema 3](#_Toc467528461)

[3. Solución del Problema 4](#_Toc467528462)

[4. Análisis de Resultados 5](#_Toc467528463)

[5. Conclusiones 6](#_Toc467528464)

[6. Bibliografía 7](#_Toc467528465)

# Introducción

Actualmente existen muchas aplicaciones de creación de rutas para evitar presas o buscar atajos y rutas que nos ahorran tiempo y gasolina, sin embargo no sabemos cómo funcionan éstas, por esta razón se realiza este proyecto, para averiguar la magia detrás de estas aplicaciones mediante grafos y algoritmos de rutas cortas.

Este proyecto trata sobre la implementación de un sistema de viajes en el lenguaje de programación C++ mediante la utilización de grafos, con este sistema se podrá viajar a través de un mapa y sus diferentes ciudades/países, los cuales son representados con puntos y conectados mediante arcos representados por líneas, a la hora de ingresar al sistema se cargará un grafo que representa esos puntos como vértices y las líneas como arcos, luego se pedirá un destino al viajero y se comenzará a recorrer la ruta más corta al destino ingresado y se le estará constantemente preguntado al viajero si desea continuar por la ruta establecida o desea calcular otra ruta. Además, este sistema debe ser capaz de permitir a dos usuarios “viajar” al mismo tiempo y llevar un historial de las rutas que los diferentes usuarios han tomado anteriormente.

También se trabajará utilizando uno de los algoritmos más populares para calcular rutas cortas: el algoritmo de Dijkstra, el cual va calculando los pesos totales de las rutas para determinar la más corta.

# Análisis del Problema

Uno de los problemas de este proyecto radica en la utilización de grafos, los cuales deben poseer ciertos datos, como por ejemplo: vértices, arcos, y las conexiones entre estos. Sin embargo, el verdadero problema reside en la necesidad de utilizar el algoritmo de dijkstra, el cual debe ser implementado para averiguar la ruta más corta hacia el destino ingresado por el usuario y el hecho de utilizar pilas de prioridades para ir guardando la ruta que va recorriendo el viajero.

Además, se debe utilizar la librería SFML de C++ para ir graficando los vértices y sus respectivos arcos y actualizar la vista de las ciudades o puntos visitados por el usuario, se debe estar preparado para recalcular la ruta si así lo desea el viajero y se deben tomar las medidas del caso.

Luego, se deben cargar grafos desde un archivo externo, especialmente el archivo que contiene el grafo que se utilizará para hacer el recorrido de los viajeros. Los cuales irán recorriendo el grafo según el destino elegido por ellos, y la ruta que irán recorriendo se irá guardando para ser mostrada/almacenada para más adelante.

# Solución del Problema

Primero, se debió instalar la librería externa y probar ciertas funcionalidades del mismo, como por ejemplo: la creación de una ventana para no trabajar en consola, la visualización del mapa, la conexión de los vértices mediante líneas, la escritura de las ciudades y luego el recorrido de los viajeros.

Para la obtención de la ruta más corta, desde un punto de inicio ingresado por el usuario a otro punto de destino, también ingresado por el usuario, se logra obtener mediante el algoritmo de ruta corta Dijkstra el cual mediante una pila de prioridad va comparando los pesos de los arcos de todos los vértices y eligiendo el arco que posea un peso total menor, marcando como visitado este punto, viéndose reflejado de inmediato el cambio en la pantalla del programa, hasta llegar a su destino. En esa pila se guarda al final la ruta más corta encontrada por el algoritmo.

Para abrir los archivos aparte se utilizaron los comandos especializados de C++ para la creación, lectura y escritura de archivos. En este proyecto se utilizaron archivos binarios, con los cuales se pueden escribir/leer estructuras de datos fácilmente, en este caso, se lee el grafo deseado, y se van ingresando los datos leídos a un pila de adyacencia, donde se van relacionando los vértices y sus arcos.

Para terminar, se creó una interfaz gráfica agradable y fácil de utilizar para los “viajeros”, donde se elige si uno o dos van a emprender viajes, en este caso, el viajero número uno será de color rojo, y el segundo será de color verde, lo cual se verá reflejado en la pantalla a la hora de iniciar el recorido.

# Análisis de Resultados

|  |  |  |
| --- | --- | --- |
| Descripción | Estado | Observaciones |
| Creación del grafo | 100 % completado |  |
| 1 o 2 jugadores | 100 % completado |  |
| Cargar grafo desde archivo | 100 % completado |  |
| Crear grafo a pie “datos quemados” | 100 % completado |  |
| Utilización del algoritmo Dijkstra para calcular la ruta más corta para llegar a un destino x | 100 % completado |  |
| El o los usuarios pueden elegir por donde ir (no ir por una ruta) | 100% completado |  |
| Utilización de una pila para los dos jugadores | 60% completado | El programa usa dos pilas, una para cada jugador (no se logró adaptar para uno) |
| Muestra en pantalla el recorrido del o de los usuario | 60% completado |  |
| Interfaz de agradable al usuario | 100 % completado |  |

# Conclusiones

Después de haber finalizado este proyecto, se entendió la importancia y el poder de los grafos y los algoritmos de ruta corta, ya sea Dijkstra, Prim o Kruskal; ya que para este tipo de problemas de mapas y tráfico son perfectos para idear una solución eficaz y eficiente a los mismo, por ejemplo en programas como WAZE o dispositivos de navegación GPS, los cuales son de gran ayuda para todos los conductores y viajeros.

Se aprendió sobre las librerías externas de C++, su utilización y sintaxis, lo cual será de gran ayuda para futuros proyectos o trabajos en este lenguaje.

Además, a lo largo de la creación de este proyecto, se fueron adquiriendo valiosos conocimientos sobre pilas, algoritmos de ruta corta y otras nuevas funciones de C++ mientras se lograba dar solución a los distintos problemas que se presentaron sobre estas estructuras de datos, lo cual será de gran importancia en el futuro cercano.

# Bibliografía

SFML. Configuración de equipo. (11 de 2016). *SFML*. Obtenido de http://www.sfml-dev.org/tutorials/2.4/

SFML, Tutotiales. (11 de 2016). Obtenido de http://www.sfml-dev.org/tutorials/2.4/start-cb.php