

Instituto Tecnológico de Costa Rica
Escuela de Computación
Curso: Estructuras de Datos

Proyecto I

La idea de este proyecto es crear una aplicación, del tipo “viewer” de mapas, que permita a un usuario visualizar de forma interactiva una serie de capas de mapas que posteriormente se puedan consultar. En general, las capacidades de la aplicación se pueden agrupar en: acercamientos y corrimientos; cargar capas de mapas, y consulta de datos.

Tipo de elementos gráficos

Un mapa consiste básicamente de tres tipos de elementos gráficos: puntos, polilíneas y polígonos. Elementos como poblados, comercios, semáforos y pozos se pueden representar como puntos (dependiendo de la escala del mapa). Ríos, caminos, carreteras, y vías de tren se pueden representar mediante polilíneas. Los polígonos se utilizan para representar países, estados, edificios, parques y plazas.

Cada capa de un mapa representa un solo tipo de elementos, es decir, no se mezclan polilíneas y polígonos, ó puntos y polígonos. Además, generalmente se asigna un color a cada capa para distinguirla de otras y aparecen etiquetadas.

Acercamientos y corrimientos

Con el fin de facilitar al usuario su trabajo con mapas muy grandes o que requieren un alto grado de detalle, se hace necesario contar con opciones que permitan realizar acercamientos (“zoom in”), alejamiento (“zoom out”) o desplazamientos (“pan”) hacia un lado u otro. Estas opciones pueden formar parte de un menú, o bien, pueden presentarse mediante botones.

Para los acercamientos se utilizará un factor fijo de “ampliación” y que consistirá en duplicar el factor de despliegue anterior. De igual forma, para el alejamiento

se utilizará un factor que represente la mitad del factor de despliegue anterior. Los acercamientos y alejamientos utilizarán como referencia el punto central de la pantalla.

En el caso de los desplazamientos, estos se podrán realizar en cuatro direcciones: arriba, abajo, izquierda o derecha. Cada desplazamiento se realizará en un 10% del tamaño total del área de dibujo utilizada. Por último es necesario contar con una opción o botón que permite “restablecer” la vista del mapa, lo cual consiste simplemente en mostrar el mapa completo en toda su extensión.

Cargar capas de mapas

Debe ser posible cargar capas de mapas en formato ASCII. Para ello se debe poder abrir un “explorar de archivos” que permita al usuario escoger el archivo que se desea cargar. Los archivos de datos a utilizar se encontrarán en formato vectorial ASCII de Grass, cuya especificación se puede encontrar en: <http://grass.osgeo.org/grass70/manuals/vectorascii.html>

Una vez que se cargue la capa se debe presentar el nombre de la misma en una lista junto con las otras capas cargadas. Se debe permitir al usuario asignar un color a la capa y modificarla posteriormente. También debe existir una opción para apagar/prender cada capa, en cuyo caso la capa debe ser desplegada u oculta dependiendo de lo que indique el usuario.

Consulta de datos

Asociado a cada archivo vectorial existirá un archivo de datos CSV (comma separated values) que incluirá una serie de datos sobre cada uno de los elementos gráficos que se describen en el archivo vectorial.

Al cargar una capa de mapa se deberá al mismo tiempo cargar su respectivo archivo de datos y asociarlos en memoria. La idea es que cuando un usuario haga click sobre un punto, polilínea o polígono en la pantalla se despliegue en una ventana (u otra zona de la pantalla) los datos asociados a ese elemento que aparecen en el archivo de datos.

Note que usted debe definir un algoritmo eficiente para detectar cual es el

elemento que el usuario está seleccionando en la pantalla. Para ello se puede utilizar el concepto de un MBB (minimal bounding box) que consiste en guardar las coordenadas mínimas y máximas de cada elemento geográfico. Además, resulta conveniente utilizar la técnica del *doble buffer* que permitirá optimizar el despliegue de la pantalla.

Consideraciones generales

1. Para almacenar los elementos geográficos y de datos en memoria debe usar estructuras de datos del tipo Lista mediante enlaces. Puede usar la implementación de Lista que aparece en el libro de texto, o desarrollar una propia. No es permitido usar la clase List que forma parte de STL u otro tipo de implementación de terceros.
2. Este proyecto debe ser desarrollado en lenguaje C++ y utilizando el ambiente de interfaces gráficas FLTK.
3. Este proyecto debe venir acompañado de la debida documentación técnica, en formato pdf. La documentación debe contar con al menos las siguientes secciones:
 - Introducción: En donde se realiza una descripción general del proyecto, explicando cuál es su objetivo y el tipo de tecnología utilizada.
 - Descripción del problema: Un resumen de la descripción mostrada en este enunciado.
 - Explicación de clases y rutinas principales: Para aquellas clases y métodos más importantes se debe realizar una explicación que permita darse una idea de la estructura general de la aplicación y la estrategia de resolución empleada.
 - Descripción del formato de archivos: Se deben explicar cada uno de los códigos utilizados en el archivo de datos para identificar cada tipo de figura, las convenciones usadas para almacenar coordenadas, y cualquier otro aspecto relevante.

- Análisis de resultados: Se deben analizar los resultados obtenidos al final del proceso de desarrollo de la aplicación. En el caso de lograr la implementación de todas las capacidades de la aplicación, se debe indicar su finalización exitosa y mencionar cualquier capacidad “extra” que se haya logrado desarrollar. En caso que no se haya podido completar la aplicación totalmente, se debe indicar cuáles capacidades no se desarrollaron y porqué, también se debe brindar alguna “estrategia” mediante la cuál se podrían haber completado dichas capacidades.
 - Conclusiones: Se deben comentar aquellos aspectos que considera positivos sobre la realización del proyecto, como también los negativos.
4. Este proyecto se puede realizar en forma individual o en parejas. No es permitido la copia de código entre grupos de estudiantes, así como incluir porciones significativas de código obtenido mediante Internet u otro medio.