UDrawingPaper

Manual Técnico

UDrawingPaper

Lenguaje Utilizado: Java Versión 17

Sistema Operativo: Windows 7 en adelante

Requisitos Adicionales: Graphviz 2

Resumen

UDrawingPaper es una aplicación de escritorio dedicada a diseñar dibujos en un estilo pixelart. La aplicación permite diversas formas de crear una imagen en función de las capas que este la conforman (similar a lo que un programa de diseño pueda manejar con las capas). Así mismo, es posible el manejo de álbumes y la visualización de la estructura que almacena la información.

El manejo de toda la aplicación depende del uso y manejo de distintas estructuras de datos. La interfaz de usuario simplemente toma la información almacenada y la muestra en pantalla.

**Árbol B:**

El árbol B es el esqueleto de toda la base de datos. Dentro de ella se encuentran los objetos de tipo cliente, los cuales son el nexo entre todas las estructuras poseyendo cada uno un árbol AVL para las imágenes, un ABB para las capas y una lista doble enlazada para el manejo de los álbumes.

La base del árbol es el nodo B. El nodo B almacena dentro de si cada uno de los clientes con los que se vaya a trabajar. Funciona también como pilar de la lista que conforma cada pagina y almacena los punteros que apuntan hacia otras páginas.

Captura de pantalla de computadora

Descripción generada automáticamente

La siguiente clase es la lista. Cada uno de los nodos del árbol está compuesto por una lista. Si bien esto pudo haberse ahorrado manejado todo desde la rama, la ventaja que se vio en manejar la lista es la separación de los métodos que se manejan en el registro. El único método que maneja la lista es añadir un nuevo nodo. Debido a que la lista maneja un orden ascendente, el método añadir se fragmenta en tres, dependiendo si la inserción la hace al inicio, en medio o al final de la lista.

Captura de pantalla de computadora

Descripción generada automáticamente

El contendedor de cada lista es la página. Como se dijo antes, la pagina pudo contener a la lista, sin embargo, los métodos que maneja la pagina son de una lógica y concepto diferente que, por facilidad, es mas conveniente manjar aparte. La pagina solo mantiene una tarea esencial: separarse cuando llegue al límite. La naturaleza del árbol B dice que al llegar a cierto numero de valores en la lista debe de separarse. Para este caso, cada que llegue a cinco se separa en dos páginas más.

El proceso es bastante simple. La regla dice que el valor intermedio de la pagina debe de subir a la lista padre y los términos que quedan se convierten en pagina derecha y pagina izquierda. El método dividir separa por nodos la lista, pone la del medio como nueva raíz; los dos primeros se unen en una lista para crear una nueva página y lo mismo con los últimos dos del lado derecho.

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

La clase Árbol es el armazón de la estructura. Mencionada, su tarea es simplemente manejar los punteros que vayan retornando los métodos. Algo que no se menciono de la pagina es que su método de división puede regresar dos cosas: un nodo o una o página. Cada vez que se regresa un nodo se asume que es un nuevo padre del árbol con todas las conexiones: Este solo se añade junto a la lista de padres que se encuentra por encima. En caso de retornar una página, lo que regresa es la pagina con el nuevo valor agregado.

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Otros métodos relevantes para mencionar son el método de búsqueda y el método niveles. El método de búsqueda, como su nombre lo indica, es el encargado de regresar el puntero con el cliente en concreto que se indica como parámetro (usado en el login). Simplemente recorre el árbol similar a como lo haría uno binario. El método de niveles se encarga de hacer un recorrido por niveles al árbol. Para eso, se utiliza la ayuda de una cola. Toda esa información es utilizada para crear un listado con los clientes que conforman la estructura.

Captura de pantalla de computadora

Descripción generada automáticamente

**Árbol AVL:**

El árbol AVL se encarga de almacenar el objeto imagen. Las imágenes tienen en su posición un identificador, el cual sirve para almacenar y manejar el árbol y un subárbol ABB de capas que maneja las imágenes almacenadas en memoria.

El nodo AVL es la base de la estructura. Al ser un árbol binario, maneja dos punteros para moverse por sus hijos izquierdo o derecho y un nodo padre utilizado en cierta parte de las rotaciones.

Captura de pantalla de computadora

Descripción generada automáticamente

De forma simple, el árbol AVL funciona como un árbol ABB. Lo que lo hace complejo es el método del balanceo. Gracias a la recursividad que posee el análisis del árbol, el enfoque usado para agregar es ir de arriba hacia abajo buscando la rama donde debe de insertarse el nodo. De ahí hacia arriba, el árbol se encarga de ir calculando alturas y comprobando si debe de realizar una rotación.

Las rotaciones no son mas que movimientos de nodos. Existen las simples y las compuestas que realizan las simples dos veces.

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

La otra función del árbol es borrar. La teoría indica que se debe de borrar el valor mas grande de la rama izquierda o la menor de la rama derecha. Se opto para la rama derecha. Para hallarlo es que existe el método grande. Este simplemente se sigue moviendo por el lado derecho del árbol hasta que encuentra un tope. Se busca el puntero que de con ese nodo y se borra.

Luego de borrar hay que revisar que el árbol se encuentre equilibrado por lo que se va de abajo hasta arriba actualizando los pesos y rotando los nodos en caso no cumplan con el equilibrio.

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

**Árbol ABB:**

El árbol binario de búsqueda se encarga de almacenar la información relacionada con las capas (matrices). Su unidad básica es el NodoABB. El nodo ABB solo posee punteros hacia sus hijos. Maneja un espacio para el nombre y uno para su contenido.

Captura de pantalla de computadora

Descripción generada automáticamente

El método de añadir del árbol ABB es bastante simple. Su funcionamiento es ir comparando su valor con el valor de la raíz y se va moviendo por las ramas hasta que encuentre que uno de los padres tiene un hijo vacío en la rama correspondiente.

Captura de pantalla de computadora

Descripción generada automáticamente

Tanto el árbol AVL como este poseen una serie de recorridos preestablecidos. Estos son el reorden, inrden y postorden. Dichos recorridos se basan en ir descendiendo por el árbol hasta que ya no pueda en cierto orden.

Captura de pantalla de computadora

Descripción generada automáticamente

**Matriz:**

La matriz es la representación gráfica del pixelart generado por la aplicación. La base de toda la estructura es un nodo matriz. Un nodo matriz es un contenedor que guarda el hexadecimal del color que se quiere guardar y posee punteros en las cuatro direcciones cardinales.

Captura de pantalla de computadora

Descripción generada automáticamente

La matriz en si solo posee un nodo raíz. La matriz posee dos llistas de cabeceras (una vertical y otra horizontal). Cada que se quiere añadir un nuevo nodo, se recorren ambas caberas y se busca si las coordenadas ingresadas existen en la lista. De no hacerlo, la aplicación comienza a recorrer las listas hasta que encuentra el espacio correspondiente. Luego ingresa todas las cabeceras que habían de por medio para mantener la sucesión.

En caso existir, lo que hace es ir a la cabecera y desde ahí recorrer esa lista para ingresarlo de tal manera que se conserve su orden ascendente. Eso mismo aplica para la otra cabecera. En caso de existir la posición, simplemente remplaza el contenido del nodo.

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

**Lista Doble Circular:**

En esta lista se almacenan los álbumes generados de los archivos de prueba. Funcionan como estructuras lineales que se mueven a la derecha o la izquierda. La cualidad especial de la lista es que guarda listas simples dentro de sus nodos.

Dentro de la aplicación, su funcionamiento se limita a la eliminación de imágenes. Cada vez que se birra una imagen, se recorre cada una de las listas en búsqueda de la imagen en cada lista.

Captura de pantalla de computadora

Descripción generada automáticamente

**Interfaz:**

El funcionamiento de la interfaz no funciona de una manera particular. La mayor parte del tiempo, el proceso consiste en obtener imágenes de las interfaces o actualizar/añadirlas. Entre las cosas que cabe destacar:

La lectura de los JSON es apoyada gracias a la librería JSONPath. Esta clase tiene la facilidad de guardar como diccionarios la información adquirida. Para ciertos casos, es conveniente porque permite castear directamente un JSONArray a un arreglo de objetos directamente.

Captura de pantalla de computadora

Descripción generada automáticamente

La generación de imágenes es parte importante para corroborar el comportamiento correcto de la aplicación, sin embargo, es imposible mantener la eficiencia de la para todos los casos. Como nota, es importante mencionar que, para algunas imágenes, el proceso de generación puede ser demasiado tardado y puede que genere problemas al intentar cargar la imagen a la aplicación.

El ultimo detalle consiste en el proceso de mover información entre ventanas. La opción seleccionada para eso fue la serialización. Se maneja un árbol B general dúrate toda la ejecución que constantemente se serializa y se deserializa conforme se van abriendo o cerrando ventanas.

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente