

Actividad Mapa de Bits

Presentado por:

Leonar Fabián González M. cód. 1150894

Nelson Andrés Sepúlveda V. cód. 1150831

Presentado a:

MSc. Marco Antonio Adarme Jaimes

Universidad Francisco de Paula Santander

Cúcuta – Norte de Santander

2014

Proposición

Una imagen en mapa de bits o imagen ráster, es una estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color, denominada matriz, que se puede visualizar en un monitor, papel u otro dispositivo de representación [1].

Ahora el mapa de bits desarrollado permite mostrar imágenes de pixeles o imágenes pixeladas de un tamaño no mayor a los 50 pixeles por 50 pixeles.

Para el desarrollo de la aplicación se hizo necesario implementar cuatro (4) clases en el lenguaje java.

La clase Pixel.java que representa un color tomado de uno de los paquetes del lenguaje. Donde allí se implementan diferentes métodos como son:

ConvertirNegro () que permite convertir un color dado a una gama de (0, 0, 0).

convertirBlanco () que permite convertir un color hasta el máximo valor del RGB (255, 255, 255).

convertirGris() que permite convertir un color a tonalidad gris, este depende de que gama de desea convertir.

La clase FilaColor.java que representa una Secuencia de pixeles donde cada posición de la secuencia almacena un pixel o color.

La clase Imagen.java que representa una Secuencia de FilaColor, ya que esta almacena objetos de tipo FilaColor que a su vez esta contiene una Secuencia de pixeles, de una forma más abstracta representaría una matriz de colores que se desea mostrar.

La clase imagen cuenta con unos métodos o funciones que permiten modificar la imagen mostrada, la clase cuenta con las siguientes funciones:

CrearMarco (): Este método permite crearle a la imagen un recuadro o marco utilizando el método de la clase Pixel.java convertirNegro () una vez invocado convierte el color a un textura negra.

InvertirGama (String tipoColorRGB): Este método recibe como parámetro un variable String el cual contiene el nombre del color a el cual se le desea invertir la gama, este color solo puede ser rojo, verde o azul. Esta función invoca a el método de convertirGris () de la clase Pixel.java. El método recorre la matriz de colores buscando las gamas pertenecientes al color

indicado. Para indicar el color deberá seleccionarlo a través de los CheckBox; los colores determinados (rojo, verde, azul) que se muestran en la figura 1.



Figura 1. Invertir Gamas.

detectarPatronGama (String tipoColorRGB): Este método recibe como parámetro una variable String el cual contiene el nombre del color a el cual se le desea detectar la gama, una vez identificados cada una de las gamas de ese color, se invoca el método `convertirBlanco ()` de la clase `Pixel.java`. De igual forma se recorre la Secuencia de `FilaColor` para identificarlos. Para seleccionar el color deberá indicarlo a través del CheckBox (ver figura 2) allí se mostrara una breve visión de gamas del color que recibe.



Figura 2. Patrones de Gama

cortar (int fila, int columna, int ancho, int alto): Este método recibe como parámetro una fila que se encuentra dentro de la Secuencia de FilaColor, también se indica la columna y así se posiciona dentro de la matriz una variable. El ancho representa la cantidad de pixeles que se desea seleccionar hacia la derecha, de igual manera el alto indica la cantidad de pixeles que se desea seleccionar hacia abajo. Una vez seleccionada toda el área a cortar se invoca el método convertirBlanco () de la clase Pixel.java. Por último la imagen se verá afectada dejando una parte totalmente blanca.

flipHorizontal (): Este método permite invertir la imagen de manera horizontal, intercambiando la primera fila con la última fila, la segunda con la antepenúltima y así sucesivamente dentro de la imagen.

CrearImagen (int ancho, int alto, String imagen): Este método permite crear la imagen que se va a tratar. El ancho representa la cantidad de filas que se desea que la imagen este conformada, el alto indica la cantidad de columnas que se desea crear, si en algún caso alto tiene un valor de cero (0) se creara un valor aleatorio para asignar un valor a la cantidad de columnas. Una vez indicados los valores de ancho y alto se crearan los objetos de la Secuencia de FilaColor insertando colores de manera aleatoria. El parámetro imagen que es de tipo String representa la dirección la imagen que se desea cargar desde el equipo para esto se utilizó la clase BufferedImage importado del paquete java.awt.image. Donde esta clase permite hacer lecturas de archivos desde la maquina también se debe indicar que tipo de archivo o archivos son los que desea abrir.

JavaDoc: El JavaDoc en formato html lo puede encontrar en el directorio llamado dist de la carpeta raíz del proyecto.