INFORME 2 PARCIAL 2 INFORMATICA II

ANDRÉS FELIPE RENDÓN VILLADA
DANIEL ANDRÉS AGÜDELO GARCÍA

Despartamento de Ingeniería Electrónica y Telecomunicaciones
Universidad de Antioquia
Medellín
Septiembre de 2021

Índice

1. Introducción

La programacion orientada a objetos (POO) representa un cambio importante en el paradigma de la programacion. Pues esta facilita en gran medida la recursividad del codigo y realizar diversos procesos que con funciones serian bastante extensos y engorrosos de realizar utilizando otros recursos de progamacion. Como por ejemplo copiar y pegar n veces un mismo codigo o crear n cantidad de funciones para un determinado proposito. Por este motivo se realiza la siguiente actividad evaluativa en la asignatura INFORMATICA II de la Universidad de Antioquia , la cual consiste en aprender a realizar el muestreo y sobremuestreo de los datos contenidos en una imagen (pixeles) , los cuales hacen posible mediante la combinación de 3 colores (rojo,verde y azul), el visualizar tonalidades de cualquier color presentes en una imagen , el objetivo de esta actividad es la implementacion de algoritmo que permita realizar el cambio en la escala de una imagen para que esta pueda ser montada en una matriz de leds de un tamaño nxn, cabe aclarar que no se permite el uso de librerias que faciliten el procesamiento de imagenes , es decir el programa presentado por los estudiantes debe ser 100 porciento implementado por nosotros los estudiantes de los programas ingenieria en telecomunicaciones e ingenieria electronica.

2. Planteamiento del problema

como se explico anterior mente en el apartado de introduccion del presente informe, el ejercicio propuesto consiste en que se nos brindara una imagen de un determinado tamaño (no necesariamente cuadrada). se debe implementar un codigo en c++ , utilizando el entorno de QTcreator, el programa implementado debe estar en la capacidad de redimensionar la imagen (alargarla o contraerla segun sea el caso), y ademas tambien debe generar un archivo .txt con la posicion de cada pixel y la representacion RGB de los pixeles de la imagen escalada , la informacion contenedia en el txt debe copiarse y luego pegarse en el codigo de la plataforma de tinkercad para simular la imagen original en el montaje de los led desarrollado previamente en dicha plataforma. adicional se debe realizar, un informe , un manual de funcionamiento y un video donde se exponga el montaje diseñado.

2.1. Ideas para la resolucion del problema

Dado que para la solución del problema no podiamos hacer uso de librerias o metodos , que facilitaran el procesamiento de imagenes , tuvimos bastantes ideas para el desarrollo de la actividad , primero pensamos en tratar la imagen como si fuera un archivo .txt para obtener la informacion contenida en esta y luego reescribir esta infomracion en otro txt usando los pixeles necesarios para visualizar la imagen en la matriz de 16x16 leds. al principio esta parecia ser la idea mas optima para la resolucion del problema pero despues de analizar mucho la descartamos. ya que no sabiamos que pixeles estabamos usando para llevarlos a la matriz. la segunda idea consistio en que debiamos encontrar el pixel de la mitad con sus respectivas coordenadas (x,y) y apartir de este pixel empezariamos a desplazarnos en diferentes direcciones hasta obtner los 256 valores asociados a los pixels que ibamos a mostrar en nuestra matriz esta idea tambien se desecho ya que si la imagen tenia algun dibujo en una esquina o fuera del area recorrida no se mostraria en la matriz de leds. la tercer idea y la que implementamos finalmente consistio en que dividiriamos el area de la imagen en pequeñas submatrinces de nxn dimensiones, estas submatrices representarian 1 pixel en la matriz de 16x16. cabe mencionar que de cada submatriz obtendriamos la combinacion rgb y luego se promediarian estos valores para obtener la combinacion de cada color en el rango de 0-255. esta fue la opcion que implementamos finalmente , ya que no importaba donde estuviese dibujado un patron en la foto , este se mostraria en la matriz sin importar el tamaño de la imagen.

2.2. Para el Montaje

para el montaje usamos la plataforma de autodesk TINKERCAD la cual nos permitio realizar la simulacion de las imagenes en la matriz realizada con tiras led de neopixel, las cuales recibian la señal de voltaje y datos por medio de un arduino para su funcionamiento. sin embargo el codigo que usariamos para realizar la simulacion se realizo en el IDE QTcreator ya que en esta plataforma no es posible realizar el manejo de archivos para obtener los datos del txt donde guardariamos las combinaciones de los valores RGB asociados a cada pixel, lo cual presenta una gran desventaja a la hora de realizar la simulacion pues constantemente toca estar abriendo el archivo txt copiando la informacion contenida en este y pegandolo en la plataforma de tinkercad para poder realizar la simulacion.

2.3. Dificultades para la solución del problema

Algunos de las dificultades que presentamos al momento de realizar la solución de la actividad , fue la poca información que encontrabamos acerca del tema , en plataformas como youtube el material que encontrabamos era bastante escaso y en su mayoria estaba en ingles por lo que muchos conceptos no quedaban bastante claros. por otra parte al momento de realizar el montaje debimos buscar la documentación de la libreria que permitia encender los leds de las tiras de neopixel y tambien para entender como se debian conectar las tiras entre si para conseguir formar la matriz , y tambien debimos estudiar los metodos asociados a la clase QImage para obtener la idea de como podriamos realizar el escalamiento de la imagen al tamaño que necesitabamos para la matriz sin utilizar estos metodos.

2.4. Citación

El video de **Ferrabacus** [1]. fue de gran ayuda para entender como podriamos realizar el modelamiento de la imagen en un tamaño mas reducido.

Tambien planeamos usar interfaz grafica usando el video de **Cunrakes** [2]. pero obtamos por hacerlo simplemente usando lineas de codigo en una aplicacion de c++

Para la parte del control de los leds usamos la información de la libreria adafruit neopixel de **ARDUINO** [3]

Referencias

[1] Ferrabacus. Programación de imágenes c ++ desde cero - 4.1. [Online]. Available: https://www.youtube.com/watch?v=HGHbcRscFsg

[2] Cunrakes. Loading a user's image on a qlabel - c++ gui with qt tutorial. [Online]. Available: https://www.youtube.com/watch?v=TxjlDTYgoqw

[3] arduino. Adafruit neopixel. [Online]. Available: https://www.arduino.cc/reference/en/libraries/adafruit-neopixel/

3. Inclusión de imágenes

En la Figura (1), representación grafica de los pixels , y como se realizaria el redimensionamiento de la imagen.

Figura 1: analizis del problema

En la Figura (2), representación grafica de los pixels formando la bandera de Colombia.

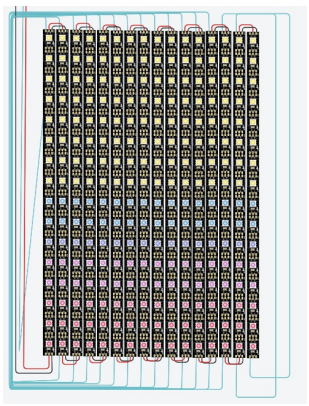

Figura 2: prueba de la matriz

En la Figura (3), representación del esquema del montaje implementado en tinkercad.

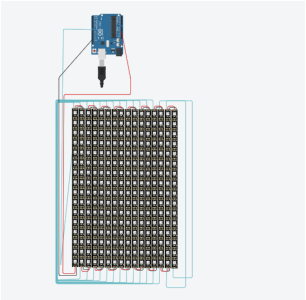

Figura 3: Montaje

En la Figura (4), representación del codigo con las clases implementado en qtcreator.

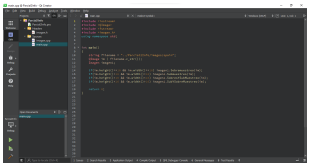

Figura 4: implmentacion qt