

Parcial2

Informe de implementación

Nombres y Apellidos del autor

Julián David Quintero Marín

Manuel Alejandro Lopez Loaiza

Informática II

Departamento de Ingeniería Electrónica y

Telecomunicaciones

Universidad de Antioquia

Medellín

September de 2021

Índice

1. Clases implementadas	2
2. Esquema de la estructura final de las clases implementadas	2
3. Módulos de código implementado donde interactúan las diferentes clases	2
4. Estructura del circuito montado	3
5. Problemas presentados durante la implementación	3

1. Clases implementadas

En este proyecto implementamos la clase QImage, con esta clase se puede almacenar una imagen al obtener su dirección y así leerla y modificarle el color de cada pixel que hace parte de la imagen. Aunque para poder modificar el color de un pixel en específico de una posición en específico, los parámetros que exige dicho método (setPixelColor) corresponde al uso de un objeto de la clase QColor, en el cual se almacenan los datos de los colores RGB que se le quieren asignar al pixel de la imagen en la posición (x,y) que se le defina. De este modo, la clase QColor también es usada dentro del proyecto.

2. Esquema de la estructura final de las clases implementadas

- Primero se incluye la clase objeto imagen.
- Luego definimos la ruta donde se cargará la imagen.
- Pedimos al usuario que añada el nombre de la imagen a la ruta.
- Se carga la Imagen deseada.
- Después se procede a hacer una comparación de pixeles para saber que método tendrá que aplicársele a la imagen cargada.
- Se le aplica el método de sub o sobre muestreo, si son iguales las dimensiones de la imagen a la de los leds sigue normal el programa.
- En los distintos métodos de redimensionamiento entrara a sus respectivos *For* con los cuales se irán guardando en un apuntador llamado colores.
- Después de que ejecutemos el programa e ingresemos el nombre de la imagen, el programa hará todo como y simplemente nos imprimirá en la pantalla la información que vamos a copiar y pegar sin la última coma en el **TINKERCAD** para mostraren los leds la imagen ingresada anteriormente.

3. Módulos de código implementado donde interactúan las diferentes clases

Modulos:

- Main.cpp
- pix.cpp
- pix.h

4. Estructura del circuito montado

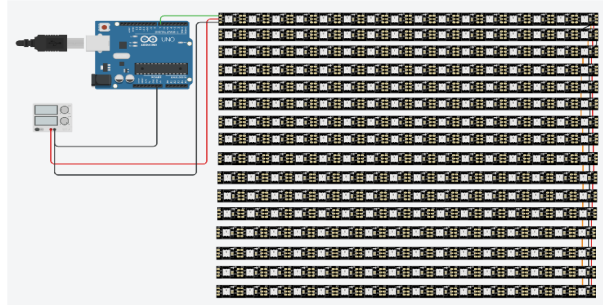


Figura 1: Montaje del circuito TINKERCAD

5. Problemas presentados durante la implementación

El mayor problema presentado fue en los métodos de submuestreo y sobremuestreo, pues al final no logramos terminarlos. En varios intentos nos salía un error que cuando el usuario ingresaba la imagen, el programa sacaba unos datos aleatorios en la primera posición y en las que le seguían, sacaba números del 1 al 499. Este error solo pasaba con las imágenes que fueran mayores o menores en las que se tendrían que usar los métodos de submuestreo o sobremuestreo según el caso.