Informe Final - Parcial II

José Alejandro Moreno Mesa - 1001369765
Miguel Ángel Restrepo Rueda - 1001017183

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

Índice

1. Introducción

Para este parcial, decidimos organizar afrontar el problema desarrollando una solución donde cargamos la imagen a través de las funciones ofrecidas por QImage, para así poder crear la imagen como objeto en primer lugar. Posteriormente, decidimos crear dos clases para realizar los métodos de sobre muestreo y submuestreo que nos permiten analizar la información del objeto imagen que habíamos ya creado para luego regresar la matriz como la necesitábamos. Finalmente, se pega en Tinkercad la información que queda guardada desde el código de Qt y se ejecuta el código para que así se pueda visualizar la información final en TinkerCad.

2. Clases implementadas

A continuación se presentan las clases implementadas como parte de la solución al reto del parcial:

2.1. finalMatrix

El entorno estará limitado, puesto que a pesar de que se le da libertad de movimiento al jugador, al ser un juego de peleas no es necesario un escenario muy grande que desmejore la experiencia del jugador, al hacerlo recorrer un escenario muy extenso que baje el dinamismo a los combates. Por otro lado, la cámara principal seguirá al jugador por lo que el entorno se estará "moviendo" a medida que el jugador se mueva.

2.2. Sobre

Esta clase es para el sobremuestreo de la imagen, es decir cuando la imagen es menor a las dimensiones de la matriz de LEDs en este caso cuando la imagen es menor a 16x16

3. Esquema de clases

Esta es la función encargada de realizar el submuestreo. Para esto, se le pide al usuario que ingrese la ruta de la imagen, con formato de imagen .jpg. Ahora para explicar la clase, consta de 16 atributos, que son los valores que se usarán posteriormente en los métodos pero no serán modificados en el código por está razón es que decidimos utilizarlos como atributos, en primera instancia está el recorrerx y recorrery el cual sirve para recorrer las columnas y filas a redimensionar respectivamente, luego se especifica el ancho de la matriz de leds, se puede coger el ancho o el largo ya que la matriz de leds (correspondiente al la matriz de neopixels montada en Tinkercad) siempre es cuadrada y se le da el ancho correspondiente que en este caso es 16 para que concuerde con el montaje de Tinkercad. Por otro lado, anchoim y altoim corresponde a el

alto y ancho de la imagen suministrada por el usuario, ahora,promedior ,promediog,promedioa, es para más adelante calcular los promedios correspondientes a cada color, redimx,redimy, indican la cantidad de filas y columnas a redimensionar respectivamente. Luego int matrixledsr[16][16], int matrixledsg[16][16],int matrixledsa[16][16], estas matrices son para mostrar los matices según sus distintos colores, sea para rojo, verde o azul.

En cuanto a los métodos, el constructor, que recibe la imagen, el método de submuestreo, la matriz ya organizada para copiar en Tinkercad, se guarda esta matriz y se escribe en el archivo de texto.

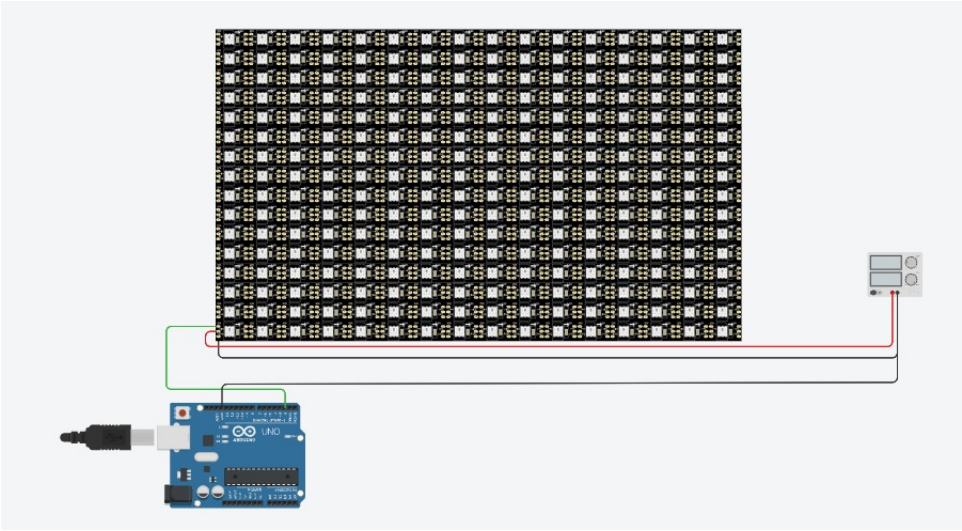
4. Modulos de codigo sobre como interactuan las clases

class sobre public: sobre(); void redim(); private: int anchoim(); int altoim(); int ancho=anchoim(); int alto=altoim();

;

5. Circuito Implementado

El circuito implementado se fundamenta en el uso de Arduino Uno y ocho tiras de NeoPixels RGB. Aquí, la conexión de las tiras de hace de la siguiente manera: El input de señal es la una tira está conectado al output de señal de la tira anterior, por lo que se tiene una continuidad en cuanto al orden de los LED. En el caso de la primera tira el input de señal está conectado al pin digital 2 de Arduino que envía la información. Se añadió una fuente de voltaje que funciona a 5v y 5mA para poder alimentar correctamente el circuito final. El voltaje está conectado directamente a la alimentación de la primera tira, y la tierra va conectada también a la primera tira y a a la tierra del Arduino para hacer el acople de tierra. Las tierras y los voltajes de cada tira están conectados todos en serie al lado izquierdo de la imagen, que es el que contiene el input de señal.



6. Problemas presentados en la implementación

Este proyecto presentó múltiples problemas, entre los que encontramos: Encontrar las técnicas correctas de sobre y submuestreo, ya que muchas de las que consideramos en un primer escenario resultaron ser incompletas o poco eficientes. Conectar el circuito, ya que al no considerar directamente la fuente externa no podíamos alimentar correctamente la matriz de NeoPixels. En el código del Arduino, nos fue difícil poder distinguir que el tipo de dato byte era el indicado para poder recibir la matriz desde Qt, ya que habíamos considerado en tipo de dato int, pero no funcionó debido al tamaño de los datos que debía almacenar. El último problema que presentó esta implementación fue que la información de la bandera salía al revés, pero con una simple modificación para hallar la "transpuesta" de la matriz y obtener los datos correctos.

7. Manual de Usuario

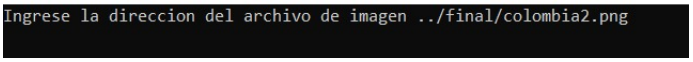
Bienvenido a los Juegos Olímpicos París 2024 Por favor siga los siguientes pasos para poder mostrar su bandera en la pantalla:

- Ingrese a Qt
- Ubíquese en el archivo "final"
- Ejecute el código
- Ingrese por consola la ubicación del archivo de imagen que desea utilizar (dirección del archivo. Mire el ejemplo para mayor claridad).

- De ser requerido por consola, ingresar el numero de veces que sea necesario nuevamente la direccion descrita en el paso anterior
- Dirigirse a la carpeta build asociada al archivo llamado “final” y abra el archivo de texto llamado “final”
- Seleccione toda la información del archivo y copiela (se recomienda usar las teclas CTRL+C)
- Diríjase al proyecto de TinkerCad llamado “Parcial 2-Info 2” en el siguiente link: https://www.tinkercad.com/things/b1QiyYciJnC-copy-of-copia-1-led-flags/editel?sharecode=XHQRJ0Oe8xZMfegVpHZVcU4hJmGIdf8O389CIuDt36U
- bra el proyecto y diríjase a la parte del código que está entre las siguientes líneas : //——— AQUI SE PEGA LA SALIDA DEL PROGRAMA DE QT———— //————————HASTA AQUI————————
- Elimine la información que haya en esa región y pegue la información que copió en el paso 6.
- Ejecute el código de TinkerCad
- Abra el Serial de la plataforma e ingrese un ”1”sin comillas ni espacios. Haga click en enter para enviarlo.
- Disfrute de ver su imagen

8. Ejemplo de Ingreso de datos

- Consola Qt



- Ingreso de direccion del archivo