Projet 1 : HSLA Images

Réalisé par : ADIL ERAAD,SAID EL OUARDI

Introduction générale

Dans le cadre de notre projet nous nous sommes intéressés à la manipulation IMAGES HSLA (HSLA :(teinte, saturation, luminance)).
A titre d'exemple le niveau de gris et illini de l'image …etc.

Transformations :

1)-lighten

```
void Image ::lighten(double amount){
    //parcourir tous mes pixles
    for(unsigned x=0;x<width();x++)
        for(unsigned y=0;y<height();y++){
            HSLAPixel &P=getPixel(x,y);
            //augumenter la luminence
            P.l+=amount;
            //ne pas depasser 1
            P.l=(P.l<1)? P.l :1;
            //ne pas depasser 0
            P.l=(P.l<0)? 0: P.l;

        }

}
```

2)-saturation

```
void Image ::saturate(double amount){
    //parcourir tous mes pixles
        //parcourir tous mes pixles
        for(unsigned x=0;x<width();x++)
            for(unsigned y=0;y<height();y++){
                HSLAPixel &P=getPixel(x,y);
                //augumenter la luminence
                P.s+=amount;
                //ne pas depasser 1
                P.s=(P.s<1)? P.s :1;
                //ne pas depasser 0
                P.s=(P.s<0)? 0: P.s;


    }
}
```

3)- `rotateColor`

```
void Image ::rotateColor(double angle){
    for(unsigned x=0;x<width();x++)
        for(unsigned y=0;y<height();y++){
            HSLAPixel &P=getPixel(x,y);
        P.h+=angle;
        //s'asurer que P.h<360
        while(P.h<360)
            P.h+=360;
        //s'asurer que P.h>360
        while(P.h>360)
            P.h-=360;

        }

}
```

4)- Grayscale

grayscale.h <Select Symbol>

```
#ifndef GRAYSCALE_H
#define GRAYSCALE_H
#include "image.h"

class Grayscale:public Image
{

public:
    //constructeur par defaut
    using Image::Image;
    Grayscale(string);
    void rz();

};

#endif // GRAYSCALE_H
```

grayscale.cpp | <Select Symbol>

```cpp
#include "grayscale.h"

Grayscale::Grayscale(string filemane):Image()
{

readFromFile(filemane);
}

void Grayscale ::rz(){
    //parcourir tous mes pixles
        //parcourir tous mes pixles
        for(unsigned x=0;x<width();x++)
            for(unsigned y=0;y<height();y++){
                HSLAPixel &P=getPixel(x,y);
                P.s=0;



    }

}
```

5)-illini

illini.cpp* | Illini::Illini(std::string, int, int) -> void

```cpp
#include "illini.h"



Illini::Illini(string filemane, int c1, int c2):Image(filemane){
    readFromFile(filemane);
   for(unsigned x=0;x<width();x++)
       for(unsigned y=0;y<height();y++){
           HSLAPixel &P=getPixel(x,y);

           if(P.h<=c2&&P.h>=c1){
                           if((c2-P.h)<(P.h-c1)){
                               P.h=c2;
                           }else{
                               P.h=c1;

                           }

}

}

}
```

illini.h — Illini(std::string, int, int) -> void

```
#ifndef ILLINI_H
#define ILLINI_H
#include"image.h"


class Illini:public Image
{

public:
    //using Image::Image;
    //Illini();
    int color1;
    int color2;

    Illini(string filemane,int color1=11,int color2=216);



   // Illini(string);
//    void ill(int color1,int color);
};

#endif // ILLINI_H
```