UEMF
الجامعة الأورومتوسطية بفاس
EUROMED UNIVERSITY OF FES
UNIVERSITÉ EUROMED DE FÈS

EIDIA
École d'Ingénierie Digitale et d'Intelligence Artificielle

Report of the project :

HSLA Images

Made by :
-Zaoui zakariae
-cheddad mohamed-ali

Supervised by :
-Pr.Belcaid

Summary :

In this project , we have implemented 4 classes following the UML class diagram below:

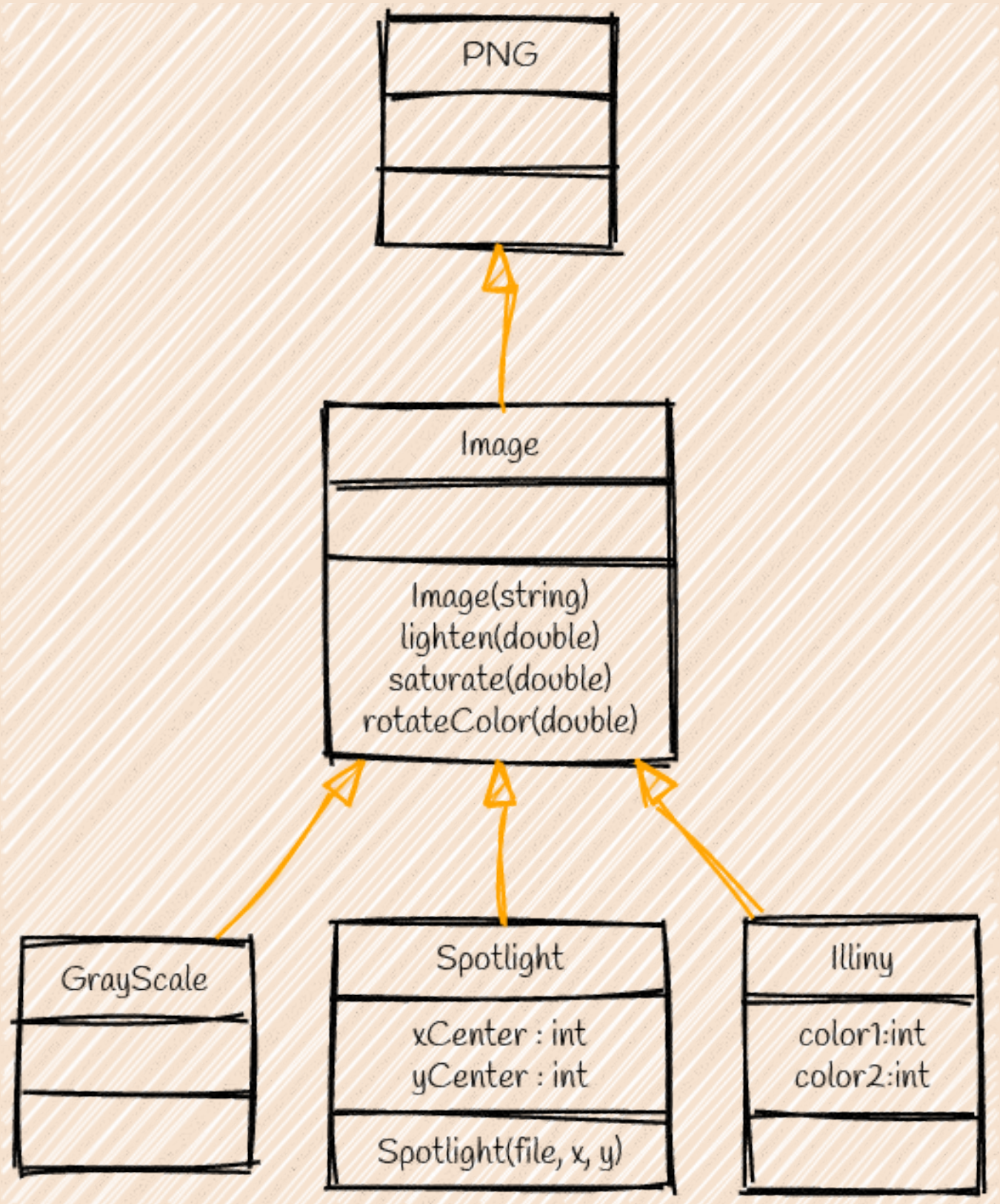


Image C:

Source Code

```
#ifndef IMAGE_H
#define IMAGE_H
#include "PNG.h"

class Image : public PNG
{
    //Additional attributs and methods
    public:
    using PNG::PNG;
    Image(string filename);
    void lighten(double amount=0.1);
    void saturate(double amount=0.1);
    void rotateColor(double angle);
};


#endif // IMAGE_H
```

image.h

Image C:

Source Code

```
void Image::lighten(double amount)
{
    for(unsigned i=0;i<width();i++)
        for(unsigned j=0;j<height();j++)
        {
            HSLAPixel &P=getPixel(i,j);
            P.l +=amount;
            P.l = (P.l>0) ? P.l :0;
            P.l = (P.l<=1) ? P.l:1;
        }
}
```

lighten method

image.cpp

Image C:

Source Code

```
void Image::saturate(double amount)
{
    for(unsigned i=0;i<width();i++)
        for(unsigned j=0;j<height();j++)
        {
            HSLAPixel &P=getPixel(i,j);
            P.s +=amount;
            P.s = (P.s>0) ? P.s :0;
            P.s = (P.s<=1) ? P.s:1;

        }

}
```

saturate method

image.cpp

Image C:

Source Code

```
void Image::rotateColor(double angle)
{
    for(unsigned i=0;i<width();i++)
        for(unsigned j=0;j<height();j++)
        {
            HSLAPixel &P=getPixel(i,j);
            P.h +=angle;
            while(P.h<0){
                P.h +=360;
            }
            while (P.h>360) {
                P.h -=360;
        }
        }

}
```

rotateColor method

image.cpp

Results:

Changing the luminance of the image using lighten(0.3) :

Input :

Output :

Results:

Changing the luminance of the image using lighten(-0.2) :

Input :

Output :

Results:

Changing the saturation of the image using saturate(0.2) :

Input :

Output :

Results:

Adding the value of an angle to the image using rotatecolor(90) :

Input :

Output :

GrayScale class :

Source code

```
#ifndef GRAYSCALE_H
#define GRAYSCALE_H
#include "image.h"
class Grayscale : public Image
{
public:
    using Image::Image;
    Grayscale(string filename);

};

#endif // GRAYSCALE_H
```

grayscale.h

GrayScale class :

Source code

```
#include "grayscale.h"
Grayscale::Grayscale(string filename):Image()
{
    readFromFile(filename);
    for(unsigned i=0;i<width();i++)
        for(unsigned j=0;j<height();j++)
        {
            HSLAPixel &P=getPixel(i,j);
            P.s=0;
        }

}
```

grayscale.cpp

Results:

Eliminating all the colors of the image :

Input :

Output :

Illini class :

Source code

```
#ifndef ILLINI_H
#define ILLINI_H
#include "image.h"
class Illini : public Image
{
public:
    using Image::Image;

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

};

#endif // ILLINI_H
```

illini.h

Illini class :

Source code

```
#include "illini.h"

Illini::Illini(string filename,int color1,int color2):Image()
{
    readFromFile(filename);int a,b;
    for(unsigned i=0;i<width();i++)
        for(unsigned j=0;j<height();j++)
        {
            HSLAPixel &P=getPixel(i,j);

            if(P.h>color1){
                if(P.h-color1<360+color1-P.h)
                    a=P.h-color1;
                else
                    a=360+color1-P.h;
            }else{
                if(-P.h+color1<360-color1+P.h)
                    a=-P.h+color1;
                else
                    a=360-color1+P.h;
            }
            if(P.h>color2){
                if(P.h-color2<360+color2-P.h)
                    b=P.h-color2;
                else
                    b=360+color2-P.h;
            }else{
                if(-P.h+color2<360-color2+P.h)
                    b=-P.h+color2;
                else
                    b=360-color2+P.h;
            }
            if(a<b)
                P.h=color1;
            else
                P.h=color2;

        }
}
```

illini.cpp

Results:

Replacing the hue of each pixel that are either the first or the second color using color1 = 11 (orange) and color2 = 216(blue).

Input :

Output :

Results:

Replacing the hue of each pixel that are either the first or the second color using color1 = 60 and color2 = 250.

Input :

Output :

Spotlight class :

Source code

```
#ifndef SPOTLIGHT_H
#define SPOTLIGHT_H
#include "image.h"
class Spotlight : public Image
{
public:
    using Image::Image;
    Spotlight(string filename,int x,int y);
};

#endif // SPOTLIGHT_H
```

spotlight.h

Spotlight class :

Source code

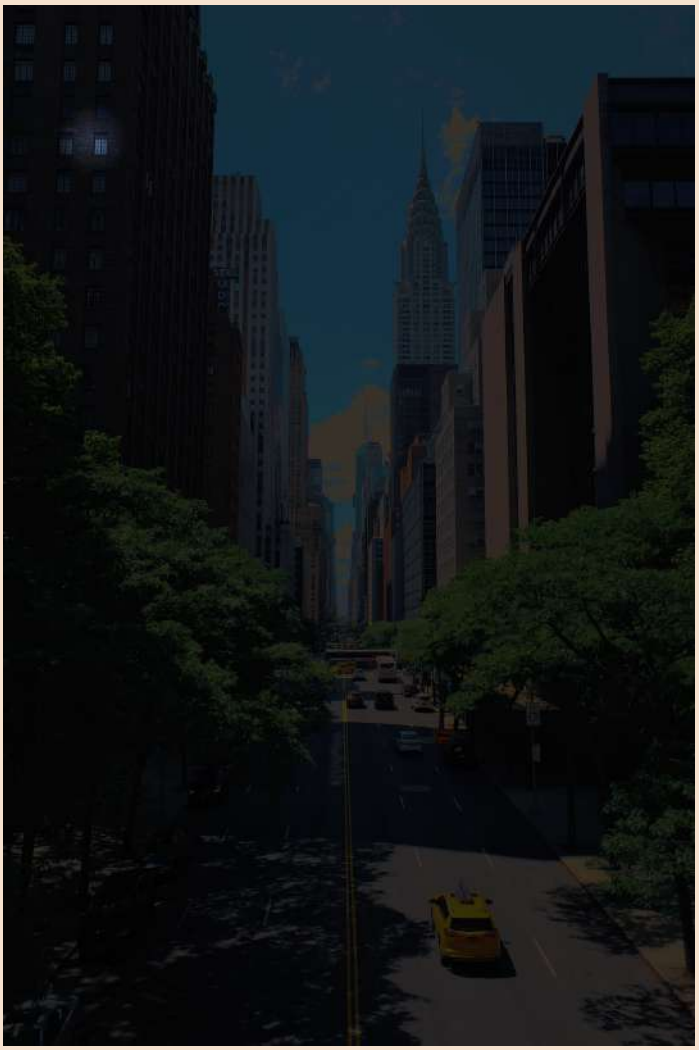
```
#include "spotlight.h"
#include <math.h>

Spotlight::Spotlight(string filename,int x,int y):Image()
{
    int a;
    readFromFile(filename);
    for(unsigned i=0;i<width();i++)
        for(unsigned j=0;j<height();j++)
        {
            HSLAPixel &P=getPixel(i,j);
            a=sqrt((x-i)*(x-i)+(y-j)*(y-j));
            if(a>160){
                P.l=0.2*P.l;
            }else{
                P.l=(1-((a*0.5)/100))*P.l;
            }
        }
}
```

spotlight.cpp

Results:

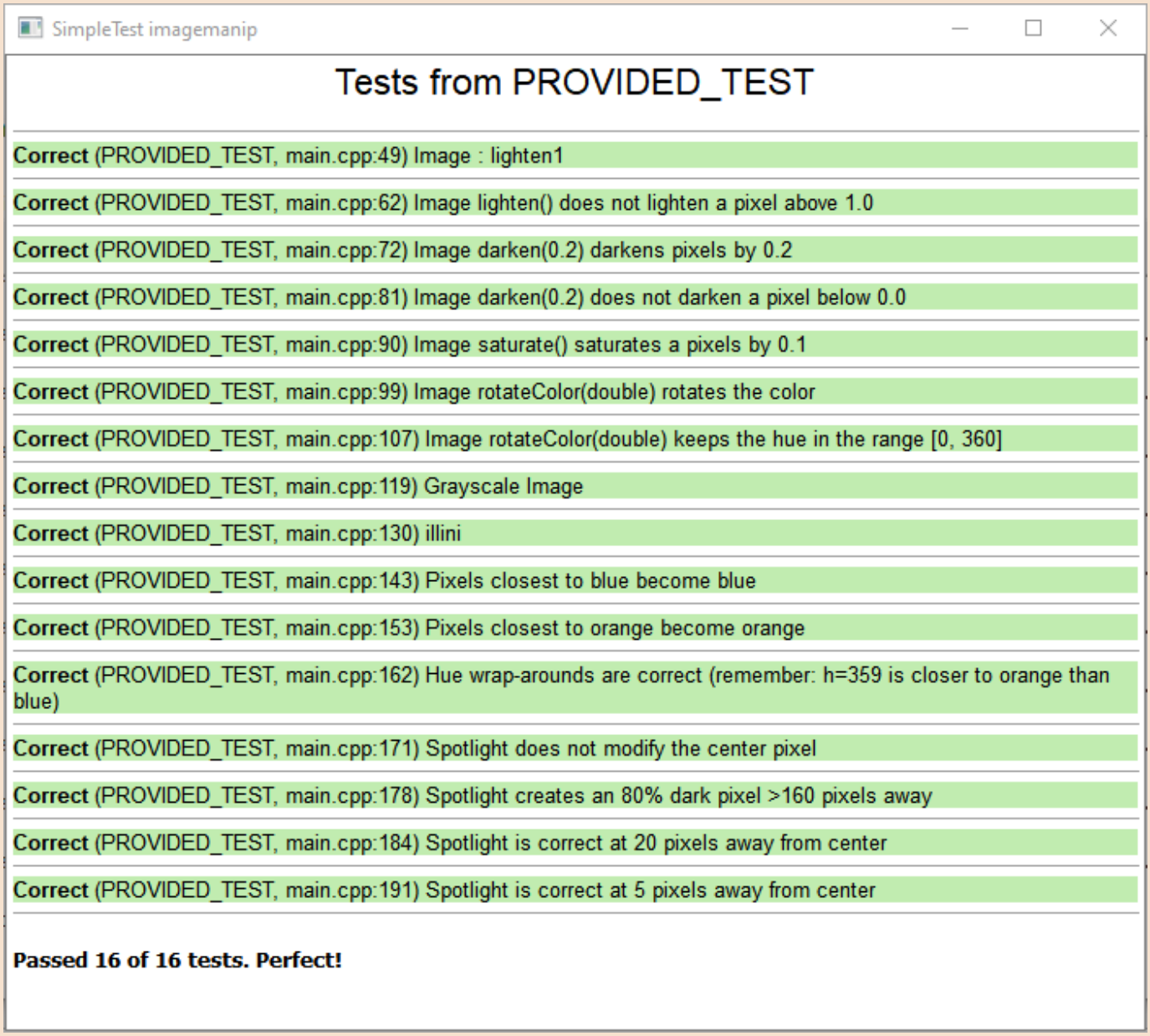
Creating a spotlight centred at the point (200,300) :



Input :

Output :

Results Of PROVIDED_TESTs :



THANK YOU!