演習課題 06 (05月17日）レポート

交換留学(文学部)　ES19-0013 ジョユンサン  
課題 6

発展課題6

*// Created by Jho on 17/05/2019.*

*// Copyright © 2019 Jho. All rights reserved.*

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include "cglec.h"

#define PI 3.14

**void** PaintCircle(Image img, **int** x0, **int** y0, **int** r, **int** g)

{

**int** x, y, xs = x0 - r, ys = y0 - r, xe = x0 + r, ye = y0 + r;

**if** (xs < 0) xs = 0;

**if** (ys < 0) ys = 0;

**if** (xe > img.Nx) xe = img.Nx;

**if** (ye > img.Ny) ye = img.Ny;

**for** (x=xs;x<xe;x++)

**for** (y = ys; y < ye; y++)

{

**if** ((x - x0) \* (x - x0) + (y - y0) \* (y - y0) <= r \* r)

{

\*(img.Data + x \* img.Ny + y) = g;

}

}

}

**void** DrawLine(Image img, **int** x1, **int** y1, **int** x2, **int** y2, **int** g)

{

**int** x, y;

**if** (x2 == x1)

{

**if** (y2 > y1) {

**for** (y = y1; y <= y2; y++) {

x = (**int**)(x1 + 0.5);

\*(img.Data + x \* img.Ny + y) = g;

}

}

**else**

**for** (y = y2; y <= y1; y++) {

x = (**int**)(x2 + 0.5);

\*(img.Data + x \* img.Ny + y) = g;

}

}

**else** {

**double** m = (**double**)(y2 - y1) / (x2 - x1);

**if** (-1 <= m && m <= 1) {

**if** (x2 > x1) {

**for** (x = x1; x <= x2; x++) {

y = (**int**)(m \* (x - x1) + y1 + 0.5);

\*(img.Data + x \* img.Ny + y) = g;

}

}

**else**

**for** (x = x2; x <= x1; x++) {

y = (**int**)(m \* (x - x2) + y2 + 0.5);

\*(img.Data + x \* img.Ny + y) = g;

}

}

**else** **if** (m > 1 || m < -1) {

**if** (y2 > y1) {

**for** (y = y1; y <= y2; y++) {

x = (**int**)((y - y1) / m + x1 + 0.5);

\*(img.Data + x \* img.Ny + y) = g;

}

}

**else**

**for** (y = y2; y <= y1; y++) {

x = (**int**)((y - y2) / m + x2 + 0.5);

\*(img.Data + x \* img.Ny + y) = g;

}

}

}

}

**int** main(**void**)

{

**int** Nx, Ny;

printf("画像の縦と横向ピクセル数は? "); scanf("%d", &Nx);

Ny = Nx;

**unsigned** **char**\* red = (**unsigned** **char**\*)malloc(**sizeof**(**unsigned** **char**) \* Nx \* Ny);

**unsigned** **char**\* green = (**unsigned** **char**\*)malloc(**sizeof**(**unsigned** **char**) \* Nx \* Ny);

**unsigned** **char**\* blue = (**unsigned** **char**\*)malloc(**sizeof**(**unsigned** **char**) \* Nx \* Ny);

**if** (red == **NULL**||green==**NULL**||blue==**NULL**)

{

printf("ERROR");

exit(0);

}

Image img\_red = { (**unsigned** **char**\*)red,Nx,Ny };

Image img\_green = { (**unsigned** **char**\*)green,Nx,Ny };

Image img\_blue= { (**unsigned** **char**\*)blue,Nx,Ny };

CglSetAll(img\_red, 0);

CglSetAll(img\_green, 0);

CglSetAll(img\_blue, 0);

**double** radian = 60 \* PI / 180;

PaintCircle(img\_blue, Nx / 2, Ny / 2, Nx/2, 255);

PaintCircle(img\_blue, Nx / 2, Ny / 2, 4 \* Nx / 10, 0);

PaintCircle(img\_green, Nx / 2, Ny / 2, 3 \* Nx / 10, 255);

PaintCircle(img\_green, Nx / 2, Ny / 2, 2 \* Nx / 10, 0);

PaintCircle(img\_red, Nx / 2, Ny / 2, Nx / 10, 255);

DrawLine(img\_red, Nx / 2 \* cos(radian) + (Nx / 2), Nx / 2 \* sin(radian) + (Nx / 2), Nx / 2 \* cos(radian) + (Nx / 2), Ny - (Nx / 2 \* sin(radian) + (Nx / 2)),255);

DrawLine(img\_green, Nx / 2 \* cos(radian) + (Nx / 2), Nx / 2 \* sin(radian) + (Nx / 2), Nx / 2 \* cos(radian) + (Nx / 2), Ny - (Nx / 2 \* sin(radian) + (Nx / 2)), 255);

DrawLine(img\_blue, Nx / 2 \* cos(radian) + (Nx / 2), Nx / 2 \* sin(radian) + (Nx / 2), Nx / 2 \* cos(radian) + (Nx / 2), Ny - (Nx / 2 \* sin(radian) + (Nx / 2)), 255);

DrawLine(img\_red, Nx / 2 \* cos(radian) + (Nx / 2), Nx / 2 \* sin(radian) + (Nx / 2), 0, Ny / 2, 255);

DrawLine(img\_green, Nx / 2 \* cos(radian) + (Nx / 2), Nx / 2 \* sin(radian) + (Nx / 2), 0, Ny / 2, 255);

DrawLine(img\_blue, Nx / 2 \* cos(radian) + (Nx / 2), Nx / 2 \* sin(radian) + (Nx / 2), 0, Ny / 2, 255);

DrawLine(img\_red, 0, Ny / 2, Nx / 2 \* cos(radian) + (Nx / 2), Nx - (Nx / 2 \* sin(radian) + (Nx / 2)), 255);

DrawLine(img\_green, 0, Ny / 2, Nx / 2 \* cos(radian) + (Nx / 2), Nx - (Nx / 2 \* sin(radian) + (Nx / 2)), 255);

DrawLine(img\_blue, 0, Ny / 2, Nx / 2 \* cos(radian) + (Nx / 2), Nx - (Nx / 2 \* sin(radian) + (Nx / 2)), 255);

DrawLine(img\_red,(Nx/2)-(Nx/2\*cos(radian)),Ny/2\*sin(radian)+(Nx/2), Nx,Ny/2, 255);

DrawLine(img\_green,(Nx/2)-(Nx/2\*cos(radian)),Ny/2\*sin(radian)+(Nx/2), Nx,Ny/2, 255);

DrawLine(img\_blue,(Nx/2)-(Nx/2\*cos(radian)),Ny/2\*sin(radian)+(Nx/2), Nx,Ny/2, 255);

DrawLine(img\_red, (Nx / 2) - (Nx / 2 \* cos(radian)), Ny / 2 \* sin(radian) + (Nx / 2), (Nx / 2) - (Nx / 2 \* cos(radian)), Nx - (Nx / 2 \* sin(radian) + (Nx / 2)), 255);

DrawLine(img\_green, (Nx / 2) - (Nx / 2 \* cos(radian)), Ny / 2 \* sin(radian) + (Nx / 2), (Nx / 2) - (Nx / 2 \* cos(radian)), Nx - (Nx / 2 \* sin(radian) + (Nx / 2)), 255);

DrawLine(img\_blue, (Nx / 2) - (Nx / 2 \* cos(radian)), Ny / 2 \* sin(radian) + (Nx / 2), (Nx / 2) - (Nx / 2 \* cos(radian)), Nx - (Nx / 2 \* sin(radian) + (Nx / 2)), 255);

DrawLine(img\_red, (Nx / 2) - (Nx / 2 \* cos(radian)), Nx - (Nx / 2 \* sin(radian) + (Nx / 2)), Nx, Ny / 2, 255);

DrawLine(img\_green, (Nx / 2) - (Nx / 2 \* cos(radian)), Nx - (Nx / 2 \* sin(radian) + (Nx / 2)), Nx, Ny / 2, 255);

DrawLine(img\_blue, (Nx / 2) - (Nx / 2 \* cos(radian)), Nx - (Nx / 2 \* sin(radian) + (Nx / 2)), Nx, Ny / 2, 255);

CglSaveColorBMP(img\_red,img\_green,img\_blue, "CircleLine.bmp");

free(red);

free(green);

free(blue);

}











