using namespace std;

#include<Windows.h>

#include<GL/glut.h>

#include<stdlib.h>

#include<math.h>

#include<conio.h>

#include<stdio.h>

#include<iostream>

#include<gl/glut.h>

void init(void);

void RenderScene(void);

void SetupRC(void);

void ChangeSize(int, int);

float cxy;

float xl = -6.0, xr = 6.0, yb = -50.0, yt = 70.0;

int xw = 900, yh = 900;

int main(int argc, char\*\* argv) {

char header[] = "Graphs of g(x)=(x^3)+(x^2)-20x, h(y)=10\*cos(y)+3, and f(x)=g(x)-h(x) by Chris Stewart";

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

cxy = ((xr - xl) / (float)xw) / ((yt - yb) / (float)yh);

cout << "calc xy" << cxy << endl;

glutInitWindowSize(900, 900);

glutInitWindowPosition(20, 20);

glutCreateWindow(header);

glutDisplayFunc(RenderScene);

SetupRC();

glutReshapeFunc(ChangeSize);

glutMainLoop();

return 0;

}

void SetupRC(void) {

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

return;

}

void RenderScene(void) {

float x, y, xdel = 0.25;

cout << "in render scene" << endl;

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(255.0, 255.0, 255.0);

cout << xw << " " << yh << " " << endl;

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glViewport(0, 0, xw, yh);

cout << "Ortho" << xl << " " << xr << " " << yb << " " << yt << endl;

glOrtho(xl, xr, yb, yt, -1.0, 1.0);

glBegin(GL\_LINES);

glVertex2f(-30, 0.0);

glVertex2f(30.0, 0.0);

for (x = -25; x <= 25.0; x += 1.0) {

glColor3f(255.0f, 255.0f, 255.0f);

glVertex2f(x, 0.0);

glVertex2f(x, 0.5);

};

glVertex2f(0.0, -50.0);

glVertex2f(0.0, 70.0);

for (y = -60.0; y <= 60.0; y += 1.0) {

glColor3f(255.0f, 255.0f, 255.0f);

glVertex2f(-0.15, y);

glVertex2f(0.15, y);

};

glEnd();

glBegin(GL\_LINE\_STRIP);

for (x = -25; x <= 25; x += xdel) {

glColor3f(0.0f, 0.0f, 100.0f);

y = (x \* x \* x) + (x \* x) - (20 \* x);

glVertex2f(x, y);

};

glEnd();

glBegin(GL\_LINE\_STRIP);

for (x = -25; x <= 25; x += xdel) {

glColor3f(255.0f, 255.0f, 0.0f);

y = 10 \* cos(x) + 3;

glVertex2f(x, y);

};

glEnd();

glBegin(GL\_LINE\_STRIP);

for (x = -30.0; x <= 30.0; x += xdel) {

glColor3f(128.0f, 0.0f, 128.0f);

y = ((x \* x \* x) + (x \* x) - (20 \* x)) - (10 \* cos(x) - 3);

glVertex2f(x, y);

};

glEnd();

glFlush();

return;

};

void ChangeSize(int w, int h) {

float ytp, xrp, wf, hf;

cout << "in ChangeSize" << w << " " << h << endl;

if (h == 0) h = 1;

wf = w; hf = h;

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glViewport(0, 0, w, h);

if (w <= h) {

ytp = (xr - xl) \* hf / (wf \* cxy) + yb;

cout << "Ortho" << xl << " " << xr << " " << yb << " " << ytp << endl;

glOrtho(xl, xr, yb, ytp, -1.0, 1.0);

}

else {

xrp = (yt - yb) \* cxy \* wf / hf + xl;

cout << "Ortho" << xl << " " << xrp << " " << yb << " " << yt << endl;

glOrtho(xl, xrp, yb, yt, -1.0, 1.0);

}

return;

}

