Nick Wright CSC328 Problem 1

Code

#include<windows.h>

#include<GL\glut.h>

#include<stdlib.h>

#include<math.h>

#include<conio.h>

#include<stdio.h>

#include <iostream>

#include <iomanip>

using namespace std;

//void init(void);//this is a function to initialize the window clear color

void RenderScene(void);//this is a function to draw a function in an opened window

void SetupRC(void);//this function sets the clear color used to set the state of the OpenGL system

//Main Program

/\*This program was designed to display 3 different functions.\*/

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

{//set up window title

char header[] = "multiple functions by Nick Wright";

// initialize the glopen utility toolkit

glutInit(&argc, argv);

// Set up the display mode with a single buffer and RGB colors

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGBA);

//Initialize window size and position

glutInitWindowSize(1016, 680);

glutInitWindowPosition(0, 0);

// Open and Label Window

glutCreateWindow(header);

//Now point to the function that will draw the scene

glutDisplayFunc(RenderScene);

// Now set the state of the rendering machine

SetupRC();

// Now execute OpenGL using RenderScene with state as set in RenderScene

glutMainLoop();

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Function SetupRC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Setup the rendering state

void SetupRC(void)

{// this function sets the clear color of an open window and clears the open window

// Set clear color to green

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

return;

}//end of SetupRC

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RenderScene Function\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void RenderScene(void)

{

double x, y, xdel = 0.25;

//clear the window

// Now clear the open window

glClear(GL\_COLOR\_BUFFER\_BIT);//note clear color was set inn SetupRC

glLoadIdentity();

//set the current drawing color to white

glColor3f(1.0, 1.0, 1.0);

//set the viewport to the window dimensions

glViewport(100, 100, 900, 500);

//Establish the clipping volumn in user units

glOrtho(-11.0, 11.0, -30.0, 55.0, 1.0, -1.0); //increased some values to allow for more to appear on the graph.

// Draw x-axis and y-axis and place tic marks on each

glBegin(GL\_LINES);

// Set end points of x-axis

glVertex2f(-8.0, 0.0);//x left

glVertex2f(8.0, 0.0);// x right

// Now put tic marks on the axis

for (x = -8.0; x <= 8.0; x += 1.0) //changed from 10 to 8

{

glVertex2f(x, 0.0);

glVertex2f(x, 0.5);

};

// Set end points of y-axis

glVertex2f(0.0, -30.0);// y bottom (changed from -10.0)

glVertex2f(0.0, 55.0);// y top (changed from 25.0)

//Now put tic marks on the axis

for (y = -30.0; y <= 55.0; y += 1.0)

{

glVertex2f(-0.15, y);

glVertex2f(0.15, y);

};

glEnd();

//setting first function draw color to red

glColor3f(1.0, 0.0, 0.0);

// Now draw the function

glBegin(GL\_LINE\_STRIP);

for (x = -6.0; x <= 5.0; x += xdel)

{

y=pow(x,3)+pow(x,2)-20\*x;

glVertex2f(x, y);

};

glEnd();

//setting first function draw color to green

glColor3f(0.0, 1.0, 0.0);

// Now draw the function

glBegin(GL\_LINE\_STRIP);

for (x = 0.0; x <= 7.0; x += xdel)//these need to be changed to x's

{

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

glVertex2f(x, y);

};

glEnd();

//setting first function draw color to blue

glColor3f(0.0, 0.0, 1.0);

// Now draw the function

glBegin(GL\_LINE\_STRIP);

for (x = -6.0; x <= 7.0; x += xdel)//these need to be changed to x's

{

y=(pow(x,3)+pow(x,2)-20\*x)-(10\*cos(x)+3);

glVertex2f(x, y);

};

glEnd();

//clear all the buffers

glFlush();

return;

};//end of render scene

Output

Histogram

Description automatically generated