**Лабораторна робота №2 – Побудова дротяних зображень з застосуванням бібліотеки OpenGL**

***Теоретичні відомості***

**Бібліотека Open GL – загальні відомості**

OpenGL – Open Graphics Library, открытая графическая библиотека. Термин "открытый" означает независимый от производителей. Имеется спецификация(стандарт) на OpenGL, где все четко задокументировано и описано. Библиотеку OpenGL может производить кто-угодно. Главное, чтобы библиотека удовлетворяла спецификации OpenGL и ряду тестов. Как следствие, в библиотеке нет никаких темных мест, секретов, недокументированных возможностей и т.п; те кто программировал под MSWindows или MSDOS, понимают о чем я говорю. Библиотеку выпускают такие корпорации, как Microsoft, Silicon Graphics, а также просто группы программистов. Одним из таких примеров служит реализация Mesa. Эту библиотеку написал целый ряд программистов, главным автором является Brian Paul. Библиотека Mesa распространяется в исходных текстах на языке Си и собирается почти для любой операционной системы. Стандарт OpenGL развивается с 1992 года. Он разрабатывается фирмой Silicon Graphics. С тех пор библиотека завоевала огромную популярность и была интегрирована со множеством языков и систем разработки приложений. Вы можете писать программу с использованием OpenGL на Си, С++, Pascal, Java и многих других языках. Основное предназначение OpenGL - программирование трехмерной графики.

**Задание:**

|  |  |
| --- | --- |
| 7. |  |

unit UnitMain;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Opengl, Vcl.ExtCtrls;

type

TFormMain = class(TForm)

PaintBox1: TPaintBox;

procedure PaintBox1Paint(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure FormDestroy(Sender: TObject);

private

{ Private declarations }

public

hrc: HGLRC;

procedure Axis;

procedure Grid;

end;

var

FormMain: TFormMain;

implementation

{$R \*.dfm}

procedure SetDCPixelFormat(hdc: HDC);

var pfd: TPixelFormatDescriptor;

nPixelFormat: integer;

begin

FillChar (pfd, SizeOf(pfd), 0);

nPixelFormat:=ChoosePixelFormat(hdc, @pfd);

SetPixelFormat(hdc, nPixelFormat, @pfd);

end;

procedure TFormMain.FormCreate(Sender: TObject);

begin

SetDCPixelFormat(Canvas.Handle);

hrc:=wglCreateContext(Canvas.Handle);

end;

procedure TFormMain.FormDestroy(Sender: TObject);

begin

wglDeleteContext(hrc);

end;

procedure TformMain.Grid();

Var I: integer; Y: single;

Begin

glColor3d(0.8,0.8,0.8);

glEnable(GL\_LINE\_STIPPLE);

glLineStipple(2, $FCFC);

Y:= 0.8;

for I := 0 to 9 do

Begin

glBegin(GL\_LINES);

glVertex2f(-1,Y);

glVertex2f(1,Y);

glVertex2f(Y,-1);

glVertex2f(Y,1);

glEnd;

Y:= Y - 0.2;

End;

glDisable(GL\_LINE\_STIPPLE);

End;

procedure TFormMain.Axis();

Var x,y,r,ps : glFloat;

n,i : Integer;

Begin

{Встановлення кольору переднього плану}

glColor3d(0,0,0);

glBegin(GL\_LINES);

// Оси

glVertex2f(-1, 0);

glVertex2f(1, 0);

glVertex2f(0, -1);

glVertex2f(0, 1);

// стрелки

glVertex2f(-0.02,0.9);

glVertex2f(0, 1);

glVertex2f(0.02,0.9);

glVertex2f(0, 1);

glVertex2f(0.93,0.035);

glVertex2f(1, 0);

glVertex2f(0.93,-0.035);

glVertex2f(1, 0);

for i := 1 to n do

begin

glVertex3f(x+r\*sin(2\*Pi \*i/n),y+r\*cos(2\*pi\*i/n),0);

end;

glEnd;

End;

procedure TFormMain.PaintBox1Paint(Sender: TObject);

var n,i: integer;

x,y: Real;

begin

{Обов’язкова команда}

wglMakeCurrent(Canvas.Handle, hrc);

{Встановлення кольору фону}

glClearColor(0.2, 0.5, 0.75, 1.0);

{Очистка екрана}

glClear(GL\_COLOR\_BUFFER\_BIT);

{Виведення на екран графіка}

glLineWidth(3);

glColor3d(1,0,0); {Червоний колір переднього плану}

glBegin(GL\_LINE\_STRIP); {Початок ламаної лінії}

x:=-1; {Абсциса початкової точки}

repeat

y:=cos(10\*x)/sin(3\*x); {Ордината поточної точки}

glVertex2d(x, y); {Додавання поточної точки як вузла до ламаної лінії}

x:=x+0.01; {Зсув абсциси поточної точки}

until x>=0;

x:=0; {Абсциса початкової точки}

repeat

y:=cos(10\*x)/sin(3\*x); {Ордината поточної точки}

glVertex2d(x, y); {Додавання поточної точки як вузла до ламаної лінії}

x:=x+0.01; {Зсув абсциси поточної точки}

until x>=1;

glEnd; {Кінець ламаної лінії}

Grid;

Axis;

{Обов’язкова команда}

wglMakeCurrent(0,0);

end;

end.

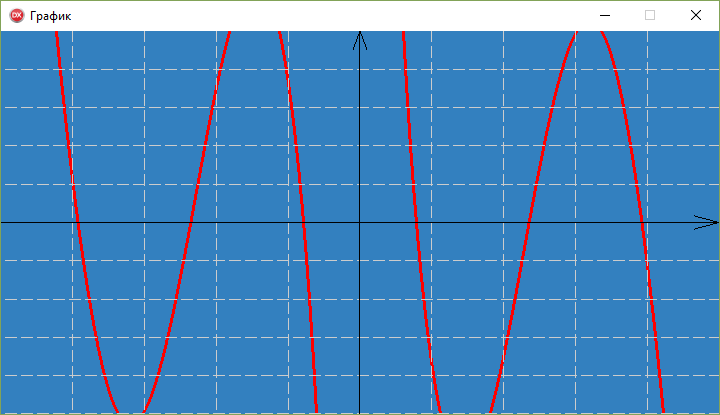


Рис. 1 “Графік”

**Вывод:** Я сформировал основные понятия программы, отработал навыки рисования объектов c использованием библиотеки Opengl, сформировал умения работать с информацией, необходимой для решения поставленной задачи.