МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники Кафедра электронных вычислительных машин

«Представление трёхмерных объектов»

Отчет

Лабораторная работа №8 по дисциплине

«Компьютерная графика»

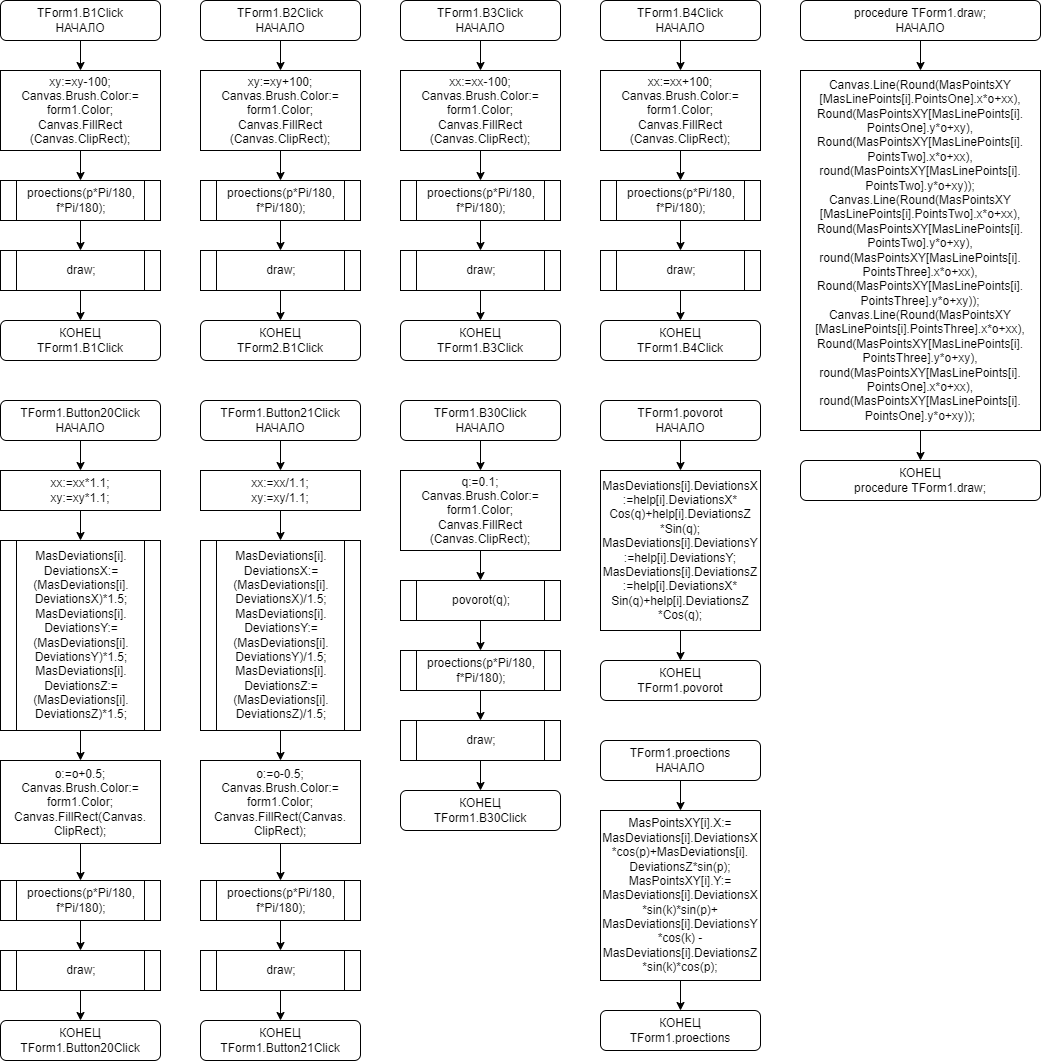
Выполнил студент группы ИВТб-21\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Коржавина А.С./

2021 г.

**Цель:** закрепить лекционный материал по теме «Представление трёхмерных объектов», изучить векторную полигональную модель.

**Схемы алгоритмов:**

****

**Исходный код**

**unit** Unit1;

{$mode objfpc}{$H+}

**interface**

**uses**

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, Menus, StdCtrls, ExtCtrls, LCLType, MMSystem;

**type**

{ TForm1 }

TXYZ = **record**

X: real;

Y: real;

Z: real;

**end**;

DeviationsXYZ = **record**

DeviationsX: real;

DeviationsY: real;

DeviationsZ: real;

**end**;

LinePoints = **record**

PointsOne: integer;

PointsTwo: integer;

PointsThree: integer;

**end**;

PointsXY = **record**

X: real;

Y: real;

**end**;

TForm1 = **class**(TForm)

B1: TButton;

B2: TButton;

B3: TButton;

B4: TButton;

Button20: TButton;

Button21: TButton;

Button30: TButton;

MainMenu1: TMainMenu;

MenuItem1: TMenuItem;

OpenDialog1: TOpenDialog;

**procedure** B1Click(Sender: TObject);

**procedure** B2Click(Sender: TObject);

**procedure** B3Click(Sender: TObject);

**procedure** B4Click(Sender: TObject);

**procedure** Button20Click(Sender: TObject);

**procedure** Button21Click(Sender: TObject);

**procedure** Button30Click(Sender: TObject);

**procedure** FormKeyDown(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**procedure** MenuItem1Click(Sender: TObject);

**procedure** proections(k, p: real);

**procedure** draw;

**procedure** helper;

**procedure** povorot(q: real);

**private**

**public**

**end**;

**var**

Form1: TForm1;

p: real = 22.208;

f: real = 20.705;

nn: integer = 1;

vp: integer = 1;

help: **array of** DeviationsXYZ;

STRCountXYZ, STRCountLine, STRCount, one, StrVarX, StrVarY, StrVarZ: string;

CountXYZ, CountLine, Count, sx, sy, b: integer;

o: real;

q: real;

xx, xy: real;

VarX, VarY, VarZ, MaxDeviations, MinDeviations: real;

MasDeviations: **array of** DeviationsXYZ;

MasDeviations2: **array of** DeviationsXYZ;

MasLinePoints: **array of** LinePoints;

MasPointsXY: **array of** PointsXY;

MasPointsXY2: **array of** PointsXY;

**implementation**

{$R \*.lfm}

{ TForm1 }

**procedure** TForm1.B1Click(Sender: TObject);

**begin**

xy := xy - 100;

Canvas.Brush.Color := form1.Color;

Canvas.FillRect(Canvas.ClipRect);

proections(p \* Pi / 180, f \* Pi / 180);

Canvas.Pen.Color := clBlack;

draw;

**end**;

**procedure** TForm1.B2Click(Sender: TObject);

**begin**

xy := xy + 100;

Canvas.Brush.Color := form1.Color;

Canvas.FillRect(Canvas.ClipRect);

proections(p \* Pi / 180, f \* Pi / 180);

Canvas.Pen.Color := clBlack;

draw;

**end**;

**procedure** TForm1.B3Click(Sender: TObject);

**begin**

xx := xx - 100;

Canvas.Brush.Color := form1.Color;

Canvas.FillRect(Canvas.ClipRect);

proections(p \* Pi / 180, f \* Pi / 180);

Canvas.Pen.Color := clBlack;

draw;

**end**;

**procedure** TForm1.B4Click(Sender: TObject);

**begin**

xx := xx + 100;

Canvas.Brush.Color := form1.Color;

Canvas.FillRect(Canvas.ClipRect);

proections(p \* Pi / 180, f \* Pi / 180);

Canvas.Pen.Color := clBlack;

draw;

**end**;

**procedure** TForm1.Button20Click(Sender: TObject);

**var**

i: integer;

**begin**

**if** o < 5.5 **then**

**begin**

xx := xx \* 1.1;

xy := xy \* 1.1;

**for** i := 0 **to** CountXYZ - 1 **do**

**begin**

MasDeviations[i].DeviationsX := (MasDeviations[i].DeviationsX) \* 1.5;

MasDeviations[i].DeviationsY := (MasDeviations[i].DeviationsY) \* 1.5;

MasDeviations[i].DeviationsZ := (MasDeviations[i].DeviationsZ) \* 1.5;

**end**;

o := o + 0.5;

Canvas.Brush.Color := form1.Color;

Canvas.FillRect(Canvas.ClipRect);

proections(p \* Pi / 180, f \* Pi / 180);

Canvas.Pen.Color := clBlack;

draw;

**end**;

**end**;

**procedure** TForm1.Button21Click(Sender: TObject);

**var**

i: integer;

**begin**

**if** o > 0.5 **then**

**begin**

xx := xx / 1.1;

xy := xy / 1.1;

**for** i := 0 **to** CountXYZ - 1 **do**

**begin**

MasDeviations[i].DeviationsX := (MasDeviations[i].DeviationsX) / 1.5;

MasDeviations[i].DeviationsY := (MasDeviations[i].DeviationsY) / 1.5;

MasDeviations[i].DeviationsZ := (MasDeviations[i].DeviationsZ) / 1.5;

**end**;

o := o - 0.5;

Canvas.Brush.Color := form1.Color;

Canvas.FillRect(Canvas.ClipRect);

proections(p \* Pi / 180, f \* Pi / 180);

Canvas.Pen.Color := clBlack;

draw;

**end**;

**end**;

**procedure** TForm1.Button30Click(Sender: TObject);

**begin**

q := 0.1;

Canvas.Brush.Color := clWhite;

Canvas.FillRect(Canvas.ClipRect);

povorot(q);

proections(p \* Pi / 180, f \* Pi / 180);

Canvas.Pen.Color := clBlack;

draw;

**end**;

**procedure** TForm1.FormKeyDown(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**begin**

**if** key = VK\_Up **then** B1.Click;

**if** key = VK\_Down **then** B2.Click;

**if** key = VK\_left **then** B3.Click;

**if** key = VK\_right **then** B4.Click;

**if** key = VK\_Add **then** button20.Click;

**if** key = VK\_Subtract **then** button21.Click;

**if** key = VK\_Space **then** button30.Click;

**end**;

**procedure** TForm1.helper;

**var**

i: longint;

**begin**

**for** i := 0 **to** countxyz - 1 **do**

**begin**

help[i].DeviationsX := MasDeviations[i].DeviationsX;

help[i].DeviationsY := MasDeviations[i].DeviationsY;

help[i].DeviationsZ := MasDeviations[i].DeviationsZ;

**end**;

**end**;

**procedure** TForm1.povorot(q: real);

**var**

i: longint;

**begin**

helper;

**for** i := 0 **to** countxyz - 1 **do**

**begin**

MasDeviations[i].DeviationsX := help[i].DeviationsX \* Cos(q) + help[i].DeviationsZ \* Sin(q);

MasDeviations[i].DeviationsY := help[i].DeviationsY;

MasDeviations[i].DeviationsZ := -help[i].DeviationsX \* Sin(q) + help[i].DeviationsZ \* Cos(q);

**end**;

**end**;

**procedure** TForm1.draw;

**var**

i: integer;

**begin**

**for** i := 0 **to** CountLine - 1 **do**

**begin**

Canvas.Line(Round(MasPointsXY[MasLinePoints[i].PointsOne].x \* o + xx), Round(MasPointsXY[MasLinePoints[i].PointsOne].y \* o + xy), Round(MasPointsXY[MasLinePoints[i].PointsTwo].x \* o + xx), round(MasPointsXY[MasLinePoints[i].PointsTwo].y \* o + xy));

Canvas.Line(Round(MasPointsXY[MasLinePoints[i].PointsTwo].x \* o + xx), Round(MasPointsXY[MasLinePoints[i].PointsTwo].y \* o + xy), round(MasPointsXY[MasLinePoints[i].PointsThree].x \* o + xx), Round(MasPointsXY[MasLinePoints[i].PointsThree].y \* o + xy));

Canvas.Line(Round(MasPointsXY[MasLinePoints[i].PointsThree].x \* o + xx), Round(MasPointsXY[MasLinePoints[i].PointsThree].y \* o + xy), round(MasPointsXY[MasLinePoints[i].PointsOne].x \* o + xx), round(MasPointsXY[MasLinePoints[i].PointsOne].y \* o + xy));

**end**;

**end**;

**procedure** TForm1.proections(k, p: real);

**var**

i: integer;

**begin**

**for** i := 0 **to** CountXYZ - 1 **do**

**begin**

MasPointsXY[i].X := MasDeviations[i].DeviationsX \* cos(p) + MasDeviations[i].DeviationsZ \* sin(p);

MasPointsXY[i].Y := MasDeviations[i].DeviationsX \* sin(k) \* sin(p) + MasDeviations[i].DeviationsY \* cos(k) - MasDeviations[i].DeviationsZ \* sin(k) \* cos(p);

**end**;

**end**;

**procedure** TForm1.MenuItem1Click(Sender: TObject);

**var**

filename: string;

txt1: text;

pp: string;

n, i: integer;

Transform: string;

**begin**

MaxDeviations := 0;

MinDeviations := 0;

**if** OpenDialog1.Execute **then begin**

filename := OpenDialog1.FileName;

AssignFile(txt1, filename);

reset(txt1);

readln(txt1, pp);

readln(txt1, Transform);

n := Pos(' ', Transform);

STRCountXYZ := Copy(Transform, 0, n - 1);

Delete(Transform, 1, n);

n := Pos(' ', Transform);

STRCountLine := Copy(Transform, 0, n - 1);

Delete(Transform, 1, n);

STRCount := Copy(Transform, 0, Transform.Length);

CountXYZ := strtoint(STRCountXYZ);

CountLine := strtoint(STRCountLine);

Count := strtoint(STRCount);

SetLength(MasDeviations, CountXYZ); SetLength(MasDeviations2, CountXYZ); SetLength(help, CountXYZ);

SetLength(MasPointsXY, CountXYZ);

**for** i := 0 **to** CountXYZ - 1 **do**

**begin**

readln(txt1, Transform);

n := Pos(' ', Transform);

StrVarX := Copy(Transform, 0, n - 1);

Delete(Transform, 1, n);

n := Pos(' ', Transform);

StrVarY := Copy(Transform, 0, n - 1);

Delete(Transform, 1, n);

StrVarZ := Copy(Transform, 0, Transform.Length);

VarX := strtoFloat(StrVarX);

VarY := strtoFloat(StrVarY);

VarZ := strtoFloat(StrVarZ);

**if** VarX >= MaxDeviations **then** MaxDeviations := VarX;

**if** VarY >= MaxDeviations **then** MaxDeviations := VarY;

**if** VarZ >= MaxDeviations **then** MaxDeviations := VarZ;

**if** VarX <= MinDeviations **then** MinDeviations := VarX;

**if** VarY <= MinDeviations **then** MinDeviations := VarY;

**if** VarZ <= MinDeviations **then** MinDeviations := VarZ;

MasDeviations[i].DeviationsX := VarX \* 300;

MasDeviations[i].DeviationsY := -VarY \* 300;

MasDeviations[i].DeviationsZ := VarZ \* 300;

**end**;

SetLength(MasLinePoints, CountLine);

**for** i := 0 **to** CountLine - 1 **do**

**begin**

readln(txt1, Transform);

n := Pos(' ', Transform);

StrVarX := Copy(Transform, 0, n - 1);

Delete(Transform, 1, n);

n := Pos(' ', Transform);

StrVarX := Copy(Transform, 0, n - 1);

Delete(Transform, 1, n);

MasLinePoints[i].PointsOne := strtoint(StrVarX);

n := Pos(' ', Transform);

StrVarY := Copy(Transform, 0, n - 1);

Delete(Transform, 1, n);

MasLinePoints[i].PointsTwo := strtoint(StrVarY);

StrVarZ := Copy(Transform, 0, Transform.Length);

MasLinePoints[i].PointsThree := strtoint(StrVarZ);

**end**;

closefile(txt1);

**end**;

xx := 960; xy := 540;

o := 3;

Canvas.Brush.Color := form1.Color;

Canvas.FillRect(Canvas.ClipRect);

proections(p \* Pi / 180, f \* Pi / 180);

Canvas.Pen.Color := clBlack;

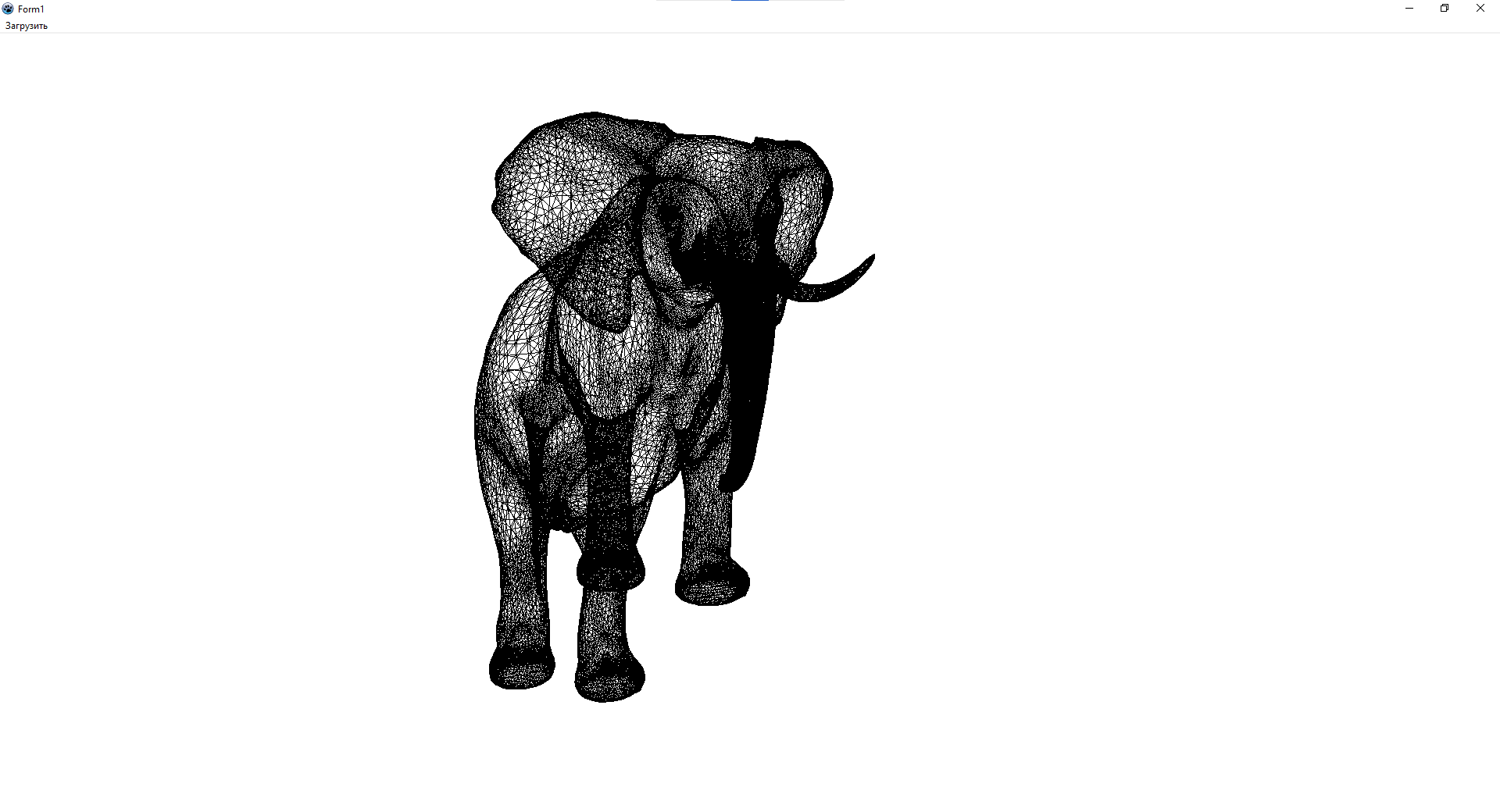
draw;

**end**;

**end**.

**Экранные формы**

****

****

**Вывод:** была реализована программа, считывающая 3д модель из файла и визуализирующая его. Также был закреплен лекционный материал по теме «Представление трёхмерных объектов» и изучена векторная полигональная модель.