**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 2**

Тема: Каркасная визуализация выпуклого многогранника. Удаление невидимых линий.

Студент: Шевчук П.В.

Группа: 80-304

Преподаватель: Чернышов Л.Н.

Дата: 23.10.2018

Оценка:

Москва, 2018

1. Постановка задачи

Разработать формат представления многогранника и процедуру его каркасной отрисовки в ортографической и изометрической проекциях. Обеспечить удаление невидимых линий и возможность пространственных поворотов и масштабирования многогранника. Обеспечить автоматическое центрирование и изменение размеров изображения при изменении размеров окна.

*Вариант 15:* восьмигранная пирамида.

1. Решение задачи

Язык программирования : C#

Используемый модуль: Drawing

Класс Graphics содержит большое количество методов, таких как DrawLine(), DrawIcon(), DrawLine(), DrawBezier(), FillElipse() или FillPolygon(). Для большинства методов класса Graphics необходимы такие примитивы, как карандаш (класс Pen), кисть (класс Brush), шрифт (Font), изображение (Bitmap) и регион (Region).

Алгоритм:

В теле программы задаются координаты пирамиды, имеющие тип класса MyPoint. В классе MyPoint объектно-ориентированным подходом переопределяем основные операции работы с векторами. В класса MyMatrix добавляем операции работы векторов.С помощью класса Triangle создаются 16 треугольников в список pyramid. Задаём размер рёбер и невидимых линий. Создаём матрицу сдвига и две матрицы поворота. Для построения пирамиды матрица сдвига умножается на матрицу масштабирования и на две матрицы поворота. В случае изометрии умножаем две заданные матрицы на две матрицы поворота, полученную матрицу умножаем на матрицу сдвига и на два.

1. Руководство по использованию программы

После запуска программы(рис. 1) в окне можно задать радиус и высоту пирамиды.

При нажатии на кнопки плюс и минус изменяется масштаб. Первая ячейка рисует невидимые линии, вторая строит пирамиду в изометрической проекции(рис.2).

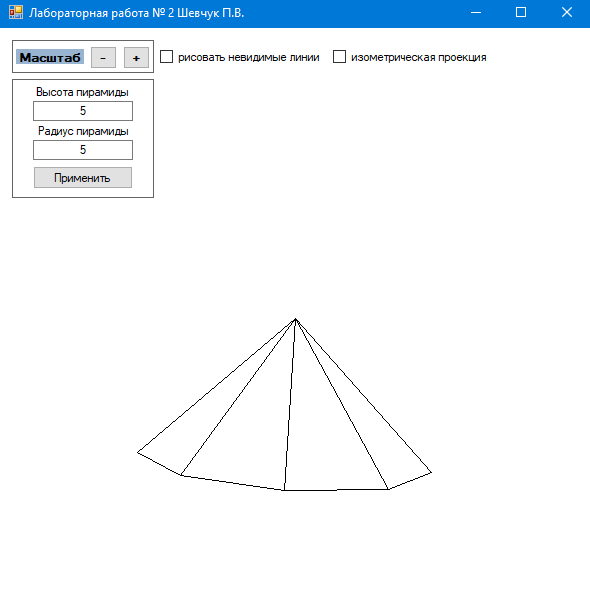


Рис.1

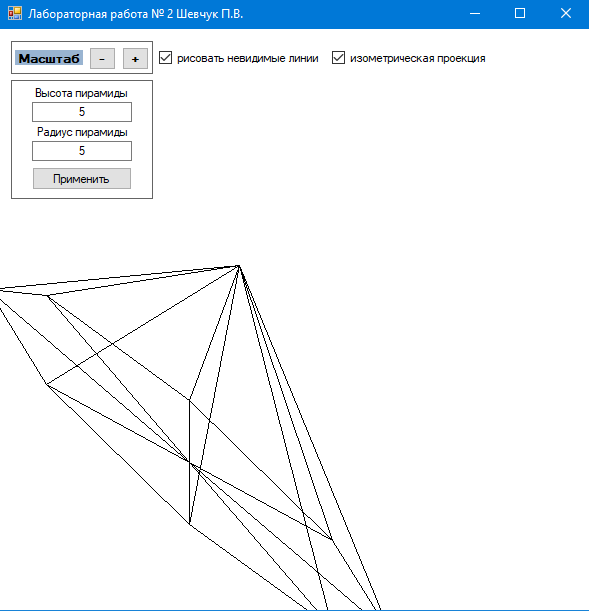


Рис.2

1. Листинг программы

/\*

Шевчук П.В.М80-304Б

Вариант 15

Каркасная визуализация восьмигранной пирамиды, обеспечить возможность пространственных поворотов и масштабирования многогранника

\*/

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

namespace lab2CG

{

public partial class Form1 : Form

{

// конструктор

public Form1()

{

InitializeComponent();

surfaces = new List<List<Triangle>>();

calcPyramid();

mx = 0;

my = 0;

cx = 0;

cy = 0;

cur = 0;

scale = 50;

mashtabK = 0;

isMouseDown = false;

}

//

public void calcPyramid()

{

double radius = double.Parse(textBoxOfPirRadius.Text);

double hight = double.Parse(textBoxOfPirHight.Text);

List<Triangle> pyramid = new List<Triangle>();

double katet = radius / Math.Sqrt(2);

// 10 точек пирамиды

MyPoint A = new MyPoint(0, 0, -hight);

MyPoint B = new MyPoint(radius, 0, -hight);

MyPoint C = new MyPoint(0, radius, -hight);

MyPoint D = new MyPoint(0, 0, 0);

MyPoint E = new MyPoint(0, -radius, -hight);

MyPoint F = new MyPoint(-radius, 0, -hight);

MyPoint M = new MyPoint(katet, katet, -hight);

MyPoint N = new MyPoint(-katet, katet, -hight);

MyPoint K = new MyPoint(-katet, -katet, -hight);

MyPoint H = new MyPoint(katet, -katet, -hight);

// 16 треугольников из пирамиды

Triangle a = new Triangle(B, M, A);

Triangle b = new Triangle(M, C, A);

Triangle c = new Triangle(C, N, A);

Triangle d = new Triangle(N, F, A);

Triangle e = new Triangle(F, K, A);

Triangle f = new Triangle(A, K, E);

Triangle g = new Triangle(A, E, H);

Triangle h = new Triangle(A, H, B);

Triangle j = new Triangle(D, M, B);

Triangle k = new Triangle(D, C, M);

Triangle l1 = new Triangle(D, N, C);

Triangle m = new Triangle(D, F, N);

Triangle n = new Triangle(D, K, F);

Triangle l = new Triangle(D, E, K);

Triangle o = new Triangle(D, H, E);

Triangle p = new Triangle(D, B, H);

// 16 треугольников добавить в список pyramid

pyramid.Add(a);

pyramid.Add(b);

pyramid.Add(c);

pyramid.Add(d);

pyramid.Add(e);

pyramid.Add(f);

pyramid.Add(g);

pyramid.Add(h);

pyramid.Add(j);

pyramid.Add(k);

pyramid.Add(l);

pyramid.Add(m);

pyramid.Add(n);

pyramid.Add(l1);

pyramid.Add(o);

pyramid.Add(p);

surfaces.Add(pyramid);

}

List<List<Triangle>> surfaces;

// текущие координаты курсора и координаты его предыдущего положения

int mx, my, cx, cy;

// индекс текущей поверхности и масштаб

int cur;

float scale;

double mashtabK;

bool isMouseDown;

// масштабирование

private void mashtabMinusButton\_Click(object sender, EventArgs e)

{

mashtabK -= 5;

Refresh();

}

private void mashtabPlusButton\_Click(object sender, EventArgs e)

{

mashtabK += 5;

Refresh();

}

// Построение графика

private void Form1\_Paint(object sender, PaintEventArgs e)

{

Pen pen = new Pen(Color.Black, 1.0f);

double zoomLevel = (scale + mashtabK) / 1000;

double coeff = Math.Max(e.ClipRectangle.Width, e.ClipRectangle.Height) \* zoomLevel;

// задаём размеры рёбер

Matrix m1 = new Matrix();

m1.m[0, 0] = Math.Sqrt(3);

m1.m[0, 1] = 0;

m1.m[0, 2] = -Math.Sqrt(3);

m1.m[1, 0] = 1;

m1.m[1, 1] = 2;

m1.m[1, 2] = 1;

m1.m[2, 0] = Math.Sqrt(2);

m1.m[2, 1] = -Math.Sqrt(2);

m1.m[2, 2] = Math.Sqrt(2);

m1.m[3, 3] = 1;

m1 = m1 \* (1 / Math.Sqrt(6));

// рёбра изометрии

Matrix m2 = new Matrix();

m2.m[0, 0] = 1;

m2.m[0, 1] = 0;

m2.m[0, 2] = 0;

m2.m[1, 0] = 0;

m2.m[1, 1] = 1;

m2.m[1, 2] = 0;

m2.m[2, 0] = 0;

m2.m[2, 1] = 0;

m2.m[2, 2] = 0;

m2.m[3, 3] = 1;

// матрица сдвига

ShiftMatrix sh = new ShiftMatrix(e.ClipRectangle.Width / 2, e.ClipRectangle.Height / 2, 0);

// матрица масштабирования

ScalingMatrix sc = new ScalingMatrix(coeff, coeff, coeff);

// создаём две матрицы поворота

RotationMatrix rtx = new RotationMatrix('X', my \* Math.PI / 180.0);

RotationMatrix rty = new RotationMatrix('Y', -mx \* Math.PI / 180.0);

Matrix tr = new Matrix();

if (izometrCheckBox.Checked) { tr = sc \* m1 \* m2 \* rtx \* rty; tr = (sh \* 2) \* tr; } // измометрия

else tr = sh \* sc \* rtx \* rty; // без изометрии

for (int i = 0; i < surfaces[cur].Count; ++i)

{

Triangle t = tr \* surfaces[cur][i];

t.Draw(pen, e.Graphics, invLinCheckBox.Checked); // невидимые линии

}

}

// управление мышкой

private void Form1\_MouseMove(object sender, MouseEventArgs e)

{

if (isMouseDown)

{

int deltaX = e.X - cx;

int deltaY = e.Y - cy;

mx += deltaX;

my += deltaY;

cx = e.X;

cy = e.Y;

Refresh();

}

}

private void Form1\_MouseDown(object sender, MouseEventArgs e)

{

isMouseDown = true;

cx = e.X;

cy = e.Y;

}

private void Form1\_MouseUp(object sender, MouseEventArgs e)

{

isMouseDown = false;

}

// Перерисовка

private void Form1\_SizeChanged(object sender, EventArgs e)

{

Refresh();

}

private void Form1\_Load(object sender, EventArgs e) { }

private void invLinCheckBox\_CheckedChanged(object sender, EventArgs e)

{

Refresh();

}

// Работа кнопки

private void buttonOfApply1\_Click(object sender, EventArgs e)

{

surfaces.Clear();

calcPyramid();

Refresh();

}

private void izometrCheckBox\_CheckedChanged(object sender, EventArgs e)

{

surfaces.Clear();

calcPyramid();

Refresh();

}

5. Выводы

Выполнив данную лабораторную работу, приобретены навыки визуализации каркасных выпуклых многогранных фигур на компьютере. Наработки, созданные в этой лабораторной работе, пригодятся в последующих.

Литература

1. Работа с graphics[Электронный ресурс].

URL: http://www.programmer-lib.ru/csharp\_page.php

2.Построение аксонометрических проекций, пирамиды [Электронный ресурс].

URL: https://verysold.wordpress.com/