Министерство образования и науки Российской Федерации

КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФИЗИКИ

КАФЕДРА РАДИОФИЗИКИ

ОТЧЕТ

По лабораторной работе №2 по предмету

Цифровая обработка изображений

Студент 4 курса,

группы 06-110

А.Ф.Хаматзянова

Казань – 2014

Постановка задачи:

Осуществить вывод на экран сцены, состоящей из 10 окружностей, случайным образом расположенных на графическом экране. Произвести отображение полученной сцены относительно вертикально оси, в качестве которой необходимо выбрать виртуальную линию раздела видеостраниц. Организовать сохранение данных видеопамяти в файл.

Код программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace programm2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public class Circle

{

public int r;

public PointF p1;

public Circle(PointF p1, int r)

{

this.p1 = p1;

this.r = r;

}

}

public PointF MatMul(PointF p1, float[,] Rotate)

{

p1.X = p1.X \* Rotate[0, 0] + p1.Y \* Rotate[0, 1];

p1.Y = p1.X \* Rotate[1, 0] + p1.Y \* Rotate[1, 1];

return p1;

}

Bitmap myBMP;

Graphics g;

public int i;

public Circle[] myCircle;

Random rn = new Random();

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

{

}

private void button1\_Click(object sender, EventArgs e)

{

myBMP = new Bitmap(pictureBox1.Width, pictureBox1.Height);

g = Graphics.FromImage(myBMP);

g.Clear(Color.White);

pictureBox1.Image = myBMP;

Pen myPen = new Pen(Color.Blue);

myCircle = new Circle[10];

for (i = 0; i < 10; i++)

{

int a = rn.Next(50, 400);

int b = rn.Next(50, 320);

int c = rn.Next(10, 50);

myCircle[i] = new Circle(new Point(a, b), c);

g.DrawEllipse(myPen, a - c, b - c, 2 \* c, 2 \* c);

}

pictureBox1.Image = myBMP;

}

private void button2\_Click(object sender, EventArgs e)

{

g = Graphics.FromImage(myBMP);

Pen myPen = new Pen(Color.Red);

float[,] Rotate = new float[2, 2];

for (i = 0; i < 10; i++)

{

Rotate[0, 0] = Math.Abs(pictureBox1.Width / myCircle[i].p1.X - 1);

Rotate[0, 1] = 0;

Rotate[1, 0] = 0;

Rotate[1, 1] = 1;

myCircle[i].p1 = MatMul(myCircle[i].p1, Rotate);

g.DrawEllipse(myPen, myCircle[i].p1.X - myCircle[i].r, myCircle[i].p1.Y - myCircle[i].r, 2 \* myCircle[i].r, 2 \* myCircle[i].r);

pictureBox1.Image = myBMP;

Array.Clear(Rotate, 0, 4);

}

Array.Clear(myCircle, 0, 10);

}

private void button3\_Click(object sender, EventArgs e)

{

myBMP = new Bitmap(pictureBox1.Width, pictureBox1.Height);

g = Graphics.FromImage(myBMP);

pictureBox1.Image = myBMP;

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

g = Graphics.FromImage(myBMP);

pictureBox1.Image = myBMP;

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

saveFileDialog1.Filter = "Jpeg Image|\*.jpg|Bitmap Image|\*.bmp|Gif Image|\*.gif";

saveFileDialog1.Title = "Save an Image File";

saveFileDialog1.ShowDialog();

if (saveFileDialog1.FileName != "")

{

System.IO.FileStream fs =

(System.IO.FileStream)saveFileDialog1.OpenFile();

switch (saveFileDialog1.FilterIndex)

{

case 1:

myBMP.Save(fs,

System.Drawing.Imaging.ImageFormat.Jpeg);

break;

case 2:

myBMP.Save(fs,

System.Drawing.Imaging.ImageFormat.Bmp);

break;

case 3:

myBMP.Save(fs,

System.Drawing.Imaging.ImageFormat.Gif);

break;

}

fs.Close();

}

}

}

}

Результат выполнения программы:

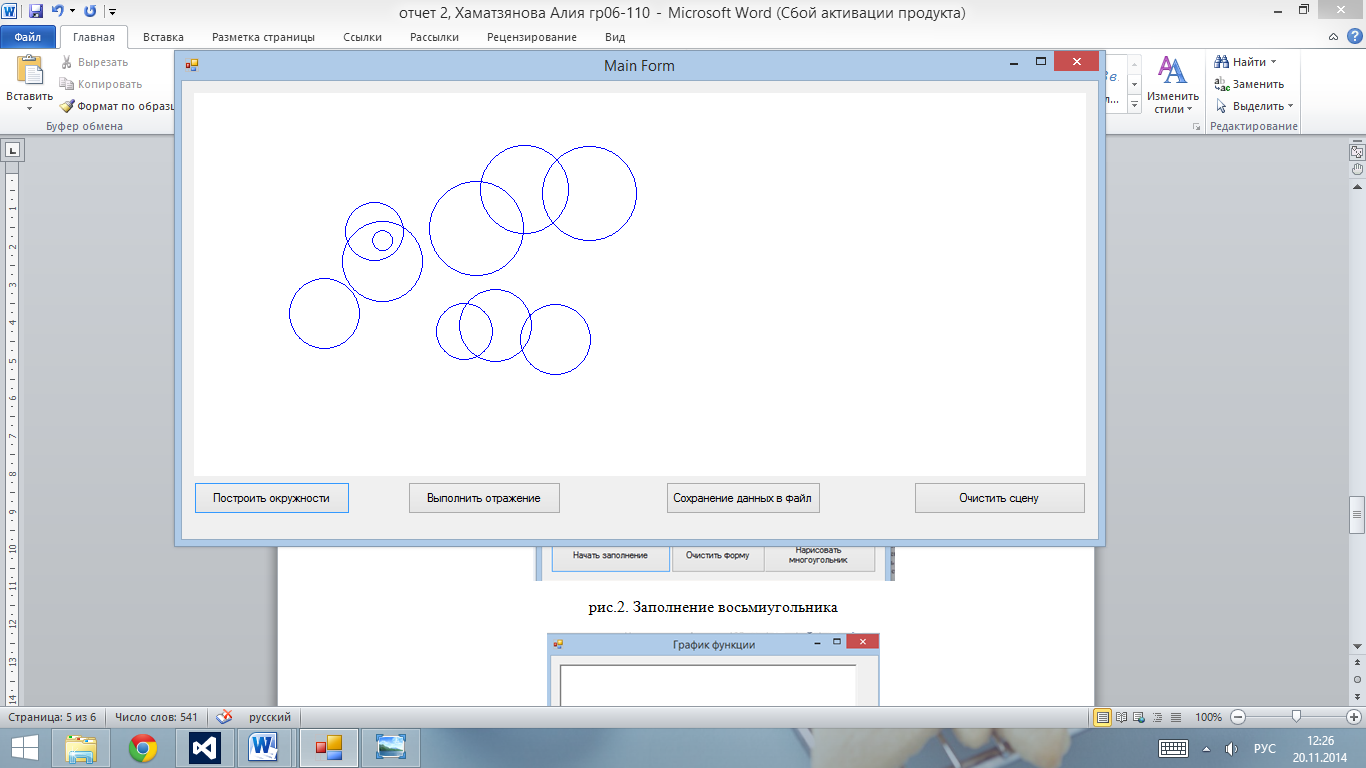


рис.1. Построение окружностей

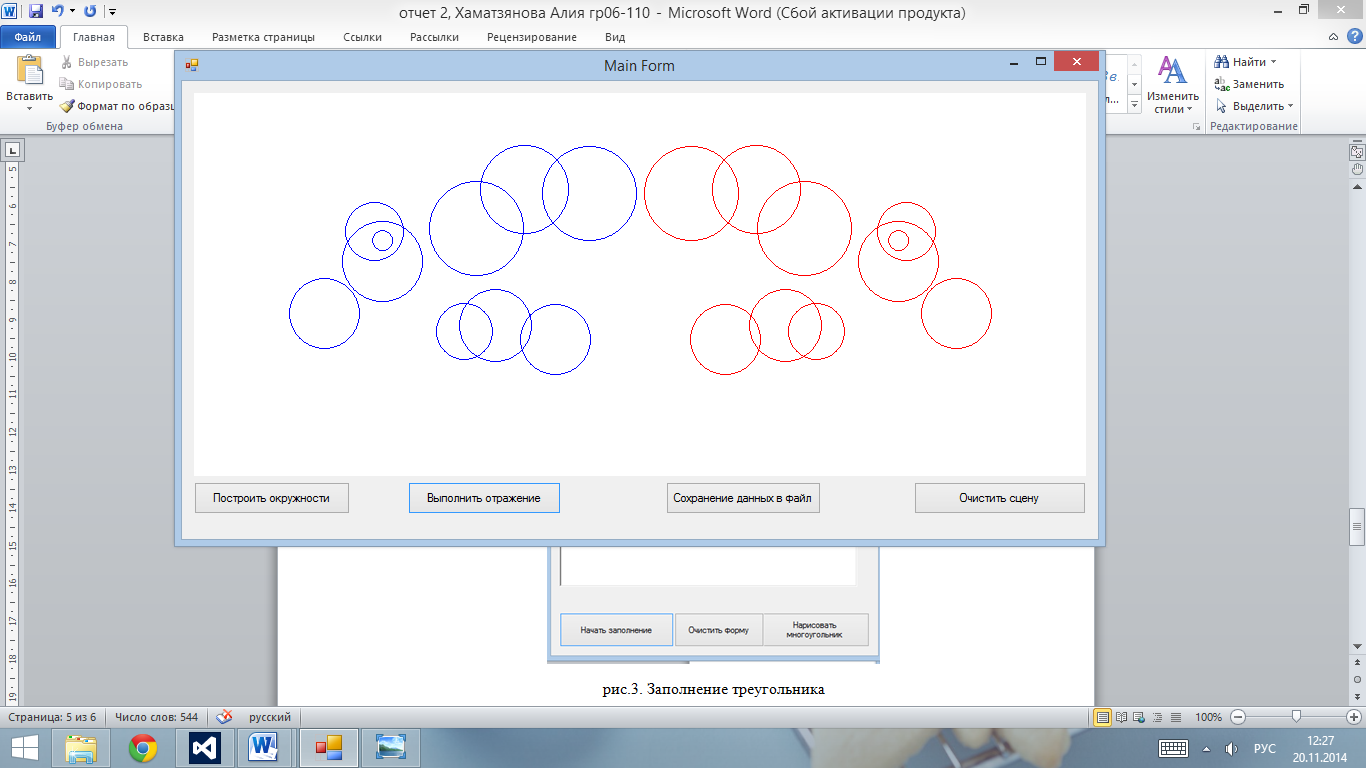


рис.2. Отображение окружностей относительно вертикальной оси

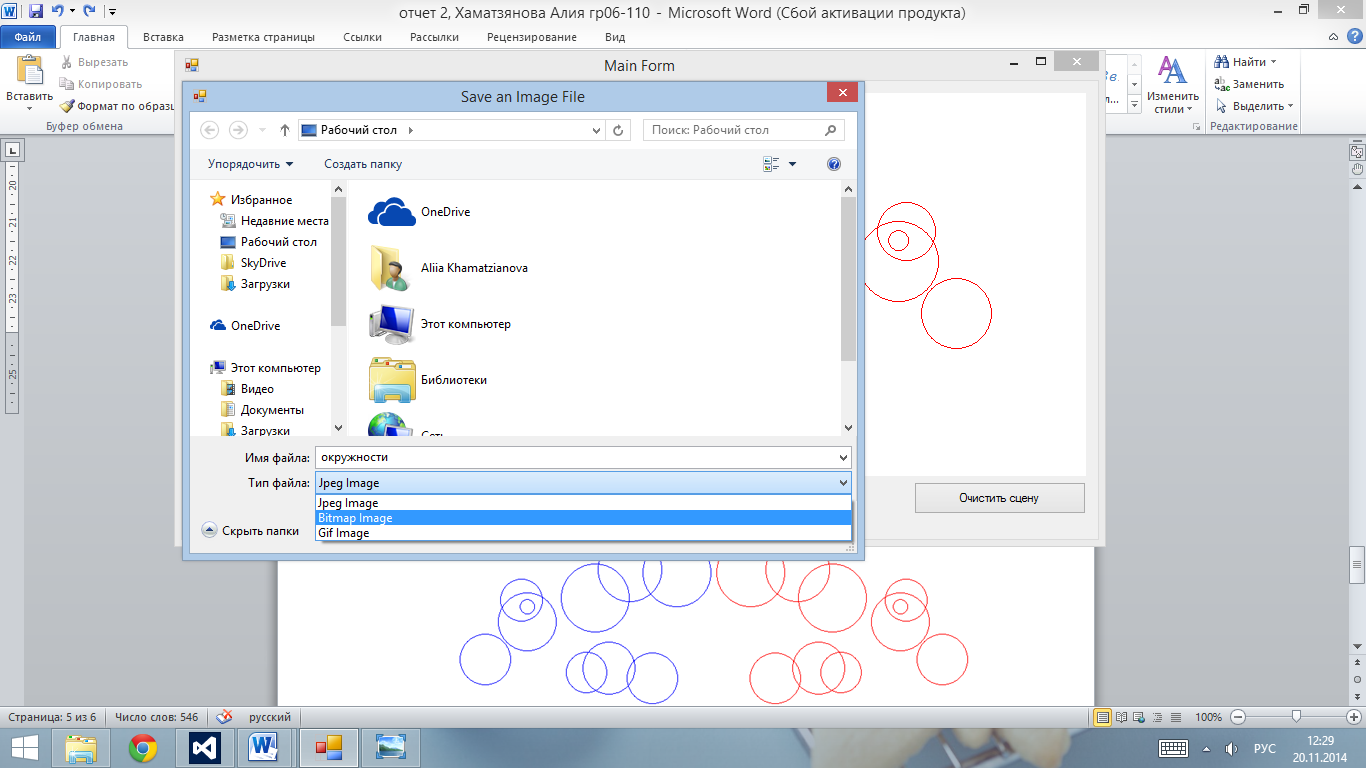


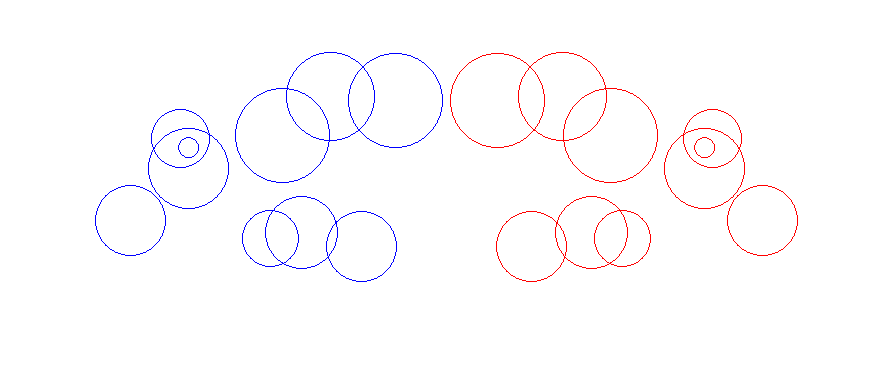
рис.3. Сохранение данных в файл

рис.4. Содержимое файла

Вывод: в результате выполнения данной лабораторной работы я изучила алгоритм отображения объектов относительно вертикальной оси. Данная программа позволяет отобразить на сцене окружности, выполнить отображение полученной сцены относительно вертикальной оси, сохранить содержимое сцены в файл, с выбором формата ( Jpeg,Bmp,Gif) , с указанием имени и расположения файла.