МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра «Математического обеспечения и применения ЭВМ»

|  |  |  |
| --- | --- | --- |
| 1002262_584459121576624_356098510_n |  | logo МОПЭВМ |
|  |  |  |

**Лабораторная работа №2**

по курсу

компьютерная графика

Вариант№ 31

Выполнили:

студент группы КТбо3-9

Митина А.Г.

Ячменев М.И.

Проверил:

Селянкин В.В.

Таганрог, 2016

**Постановка задачи**

Задание заключается в разработке программы развертки прямой и окружности

**Вариант задания**

Необходимо написать программу развертки прямой и окружности в 5 октане.

**Описание алгоритм**

Для развертки прямой в 5 октане необходимо использовать декремент по x и инкремент по y на каждом шаге. При этом дельта x и y вычисляется, как разница между конечной и начальной точкой по x и y соответственно.

Для развертки окружности в 5 октане по алгоритму Брезенхема мы используем декремент по y каждом шаге, при этом пока значение x больше значения y выполняем следующее:

Если d положительное, то вычисляем d = d + 4 \* Y + 6 иначе d = d + 4 \* (Y -X) + 10; и инкрементируем x

**Описание программы**

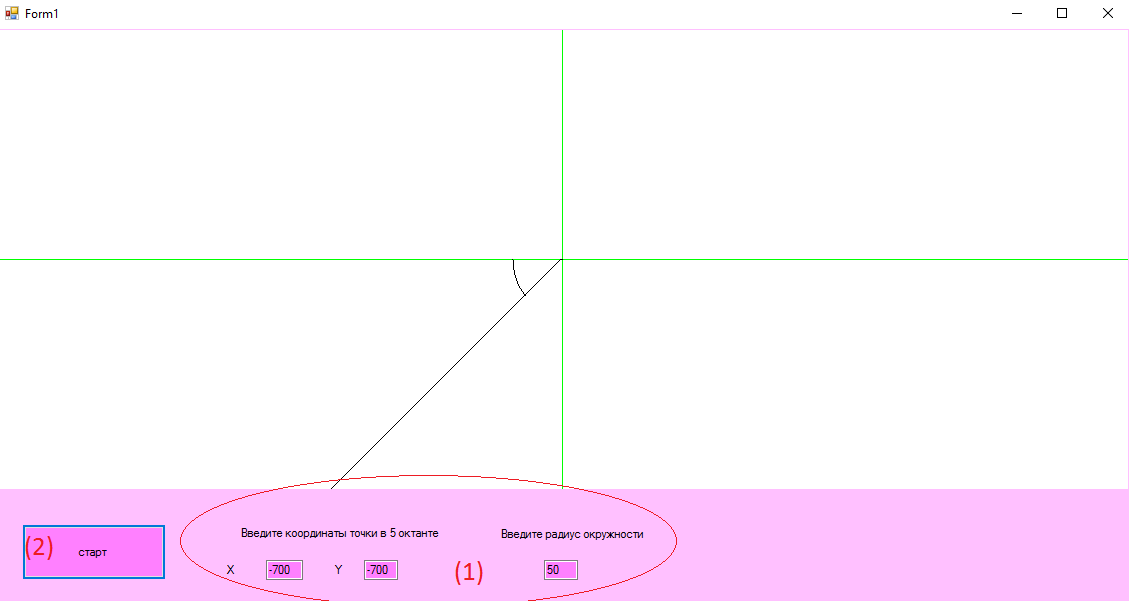
Входные значения: координаты точек отрезка в декартовой системе координат и радиус окружности .

Выходные значения: растровая развертка отрезка и окружности на экране, либо сообщение о не корректных входных данных.

Системные требования: процессор Intel Core I3 2.0Ггц

Видеокарта Nvidia GeForce 940M

Оперативная память 8 Гб

В форме (1) пользователь вводит необходимые значения. Эти значения обрабатываются по нажатию кнопки (2). Если данные введены некорректно, то появляется соответствующее сообщение(3).****

****

В методе BackGround() происходит задание фона и системы координат. В мтоде button1\_Click() происходит проверка вводимой информации, а именно проверка на принадлежность точек прямой 5 октану и положительному радиусу окружности. В методе SetArc() и SetLine рисуется окружность и прямая соответственно. Методы GetX и GetY возвращают значения по x и по yв пересчете на новые координаты.

**Листинг**

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;

using System.Windows;

namespace CG\_2

{

public partial class Form1 : Form

{

Graphics gr; //We declare an object - graphics, which will draw

Pen p = new Pen(Color.Lime); //We declare an object - a pencil, which will draw the contour

SolidBrush fon; //We declare an object - fill, to fill the background respectively

public Form1()

{

InitializeComponent();

}

private void BackGround()

{

Color pix = Color.Chocolate;

gr = pictureBox1.CreateGraphics(); //initialize an object of type Graphics

fon = new SolidBrush(Color.Black);

gr.FillRectangle(fon, 0, 0, pictureBox1.Width, pictureBox1.Height); // tied to a PictureBox

gr.DrawLine(p, 0, pictureBox1.Height / 2, pictureBox1.Width, pictureBox1.Height / 2);

gr.DrawLine(p, pictureBox1.Width / 2, 0, pictureBox1.Width / 2, pictureBox1.Height);

}

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

{

BackGround();

Point Start\_Coor = new Point(0, 0);

Point Point\_Coor = new Point(Int32.Parse(X\_Box.Text), Int32.Parse(Y\_Box.Text));

int r = Int32.Parse(RBox.Text);

if ((Point\_Coor.X < 0) && (Point\_Coor.Y >= Point\_Coor.X) && (Point\_Coor.Y < 0) && r >= 0)

{

SetLine(Start\_Coor, Point\_Coor);

SetArc(r);

}

else

{

label4.Text = "Некорректные данные!";

}

}

private void SetArc(int r)

{

Point start = new Point(-r, 0);

int d = 3 - 2 \* r;

while(start.X < start.Y)

{

PutPixel(GetX(start.X), GetY(start.Y));

if(d > 0)

{

d = d + 4 \* start.Y + 6;

}

else

{

d = d + 4 \* (start.Y - start.X) + 10;

start.X++;

}

start.Y--;

}

}

private void SetLine( Point start, Point finish)

{

int x = start.X;

int y = start.Y;

int dx = finish.X - start.X;

int dy = finish.Y - start.Y;

int D = -dx;

int DX = dx >> 1;

int DY = dy >> 1;

while (x > finish.X)

{

PutPixel(GetX(x), GetY(y));

--x;

D += DY;

if (D < 0)

{

--y;

D -= DX;

}

}

}

private void PutPixel(int x,int y)

{

gr.FillRectangle(Brushes.Azure, x, y, 1, 1);

}

private int GetX(int X)

{

return X + pictureBox1.Width/2;

}

private int GetY(int Y)

{

return pictureBox1.Height/2-Y;

}

private void label3\_Click(object sender, EventArgs e)

{

}

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

{

}

}

}