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 меньше значения y. Определяем в каком из 3 направлений следует идти.

**Описание программы**

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

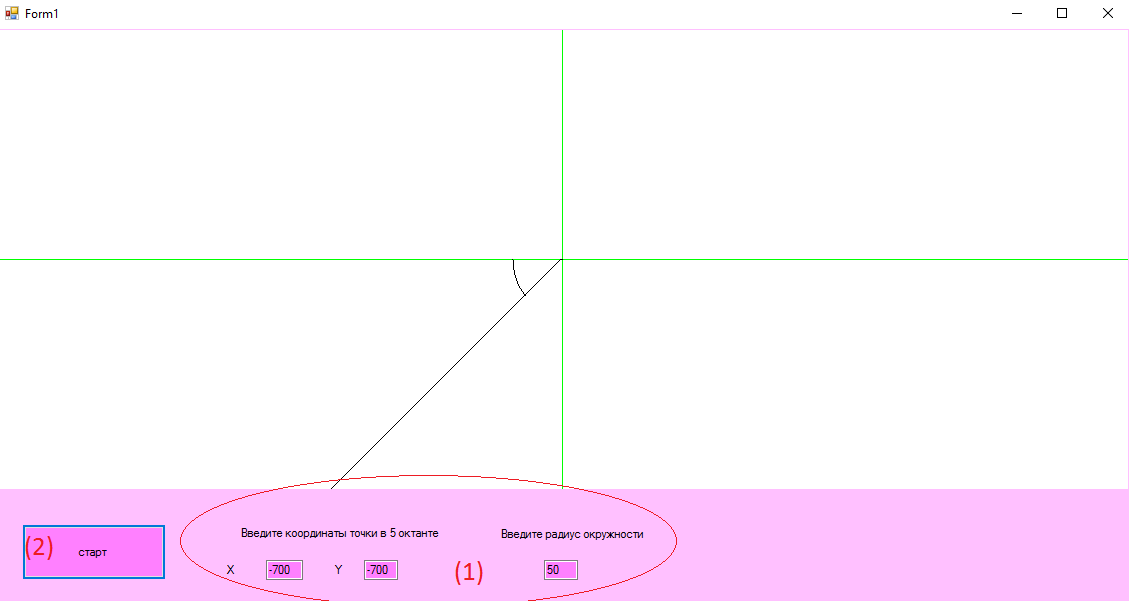
Выходные значения: растровая развертка отрезка и окружности на экране, либо сообщение о не корректных входных данных.

Системные требования: процессор Intel Core I3 2.0Ггц

Видеокарта Nvidia GeForce 940M

Оперативная память 8 Гб

Программы написана на языке С# в среде разработке Visual Studio 2015.

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**

1. using System;
2. using System.Collections.Generic;
3. using System.ComponentModel;
4. using System.Data;
5. using System.Drawing;
6. using System.Linq;
7. using System.Text;
8. using System.Threading.Tasks;
9. using System.Windows.Forms;
10. using System.Windows;
11. namespace CG\_2
12. {
13. public partial class Form1 : Form
14. {
15. Graphics gr;
16. Pen p = new Pen(Color.Lime);
17. SolidBrush fon;
19. public Form1()
20. {
21. InitializeComponent();
22. }
23. *// рисуем осикоординат*
24. private void BackGround()
25. {
26. Color pix = Color.Chocolate;
27. gr = pictureBox1.CreateGraphics();
28. fon = new SolidBrush(Color.Black);
30. gr.FillRectangle(fon, 0, 0, pictureBox1.Width, pictureBox1.Height);
31. gr.DrawLine(p, 0, pictureBox1.Height / 2, pictureBox1.Width, pictureBox1.Height / 2);
32. gr.DrawLine(p, pictureBox1.Width / 2, 0, pictureBox1.Width / 2, pictureBox1.Height);
33. }
34. *// обрабатываем входные данные*
35. private void button1\_Click(object sender, EventArgs e)
36. {
37. BackGround();
38. if (IsRightData())
39. {
40. Point Start\_Coor = new Point(0, 0);
41. Point Point\_Coor = new Point(Int32.Parse(X\_Box.Text), Int32.Parse(Y\_Box.Text));
42. int r = Int32.Parse(RBox.Text);
43. if ((Point\_Coor.X < 0) && (Point\_Coor.Y >= Point\_Coor.X) && (Point\_Coor.Y < 0) && r >= 0)*// проверка на корректность*
44. {
45. label4.Text = "";
46. SetLine(Start\_Coor, Point\_Coor);
47. int x = Int32.Parse(sentreX\_Box.Text);
48. int y = Int32.Parse(sentreY\_Box.Text);
49. SetArc(r, x, y);
50. }
51. else
52. {
53. label4.Text = "Некорректные данные!";
54. }
55. }
56. }
57. private bool IsRightData()
58. {
59. int result;
60. bool ret = **true**;
61. if (String.IsNullOrEmpty(X\_Box.Text) || !Int32.TryParse(X\_Box.Text, out result))
62. {
63. ans\_line.Text = "Некорректный Х";
64. ret = **false**;
65. }
66. else
67. {
68. ans\_line.Text = "OK";
69. }
70. if (String.IsNullOrEmpty(Y\_Box.Text) || !Int32.TryParse(Y\_Box.Text, out result))
71. {
72. ans\_line.Text = "Некорректный Y";
73. ret = **false**;
74. }
76. if (String.IsNullOrEmpty(RBox.Text) || !Int32.TryParse(RBox.Text, out result))
77. {
78. label3.Text = "Некорректный радиус";
79. ret = **false**;
80. }
81. else
82. {
83. label3.Text = "OK";
84. }
85. if (String.IsNullOrEmpty(sentreX\_Box.Text) || !Int32.TryParse(sentreX\_Box.Text, out result))
86. {
87. label5.Text = "Некорректный Х";
88. ret = **false**;
89. }
90. else
91. {
92. label5.Text = "OK";
93. }
94. if (String.IsNullOrEmpty(sentreY\_Box.Text) || !Int32.TryParse(sentreY\_Box.Text, out result))
95. {
96. label5.Text = "Некорректный Y";
97. ret = **false**;
98. }
100. return ret;
101. }
102. *// равертка окружности по алгоритму Брезенхема*
103. private void SetArc(int radius, int \_x, int \_y)
104. {
105. int x = -radius, y = 0, gap = 0, delta = (2 - 2 \* radius);
106. while (x <= 0 && x <= y)
107. {
108. PutPixel(GetX(x + \_x), GetY(y + \_y));
109. PutPixel(GetX(x + \_x), GetY(-y + \_y));
110. PutPixel(GetX(-x + \_x), GetY(-y + \_y));
111. PutPixel(GetX(-x + \_x), GetY(y + \_y));
112. PutPixel(GetX(y + \_x), GetY(x + \_y));
113. PutPixel(GetX(-y + \_x), GetY(x + \_y));
114. PutPixel(GetX(-y + \_x), GetY(-x + \_y));
115. PutPixel(GetX(y + \_x), GetY(-x + \_y));
117. gap = 2 \* (delta + y) - 1;
118. if (delta < 0 && gap <= 0)
119. {
120. y--;
121. delta -= 2 \* y + 1;
122. continue;
123. }
124. if (delta > 0 && gap > 0)
125. {
126. x++;
127. delta += 2 \* x + 1;
128. continue;
129. }
130. x++;
131. delta += 2 \* (x - y);
132. y--;
133. }
135. }
136. *// развертка прямой*
137. private void SetLine( Point start, Point finish)
138. {
139. int x = start.X;
140. int y = start.Y;
141. int dx = finish.X - start.X;
142. int dy = finish.Y - start.Y;
143. int D = -dx;
144. int DX = dx >> 1;
145. int DY = dy >> 1;
146. while (x > finish.X)
147. {
148. PutPixel(GetX(x), GetY(y));
149. --x;
150. D += DY;
151. if (D < 0)
152. {
153. --y;
154. D -= DX;
155. }
156. }
157. }
158. private void PutPixel(int x,int y)
159. {
160. gr.FillRectangle(Brushes.Aqua, x, y, 1, 1);
161. }
162. *// вернуть координаты по x в пересчете на декартовую систему*
163. private int GetX(int X)
164. {
165. return X + pictureBox1.Width/2;
166. }
167. *// вернкть координы по y*
168. private int GetY(int Y)
169. {
170. return pictureBox1.Height/2-Y;
171. }
173. private void label3\_Click(object sender, EventArgs e)
174. {
176. }
178. private void pictureBox1\_Click(object sender, EventArgs e)
179. {
181. }
183. private void label5\_Click(object sender, EventArgs e)
184. {
186. }
187. }
188. }