1. **Анимация**

Задание 1. анимации секундной стрелки часов

Листинг программы:

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Task\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private int x1, y1, x2, y2, r;

private double a;

private Pen pen = new Pen(Color.DarkRed, 2);

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

{

Graphics g = e.Graphics;

g.DrawLine(pen, x1, y1, x2, y2);

}

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

{

r = 150;

a = 0;

x1 = ClientSize.Width / 2;

y1 = ClientSize.Height / 2;

x2 = x1 + (int)(r \* Math.Cos(a));

x2 = y1 - (int)(r \* Math.Sin(a));

}

private void timer1\_Tick(object sender, EventArgs e)

{

a -= 0.1;

x2 = x1 + (int)(r \* Math.Cos(a));

x2 = y1 - (int)(r \* Math.Sin(a));

Invalidate();

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

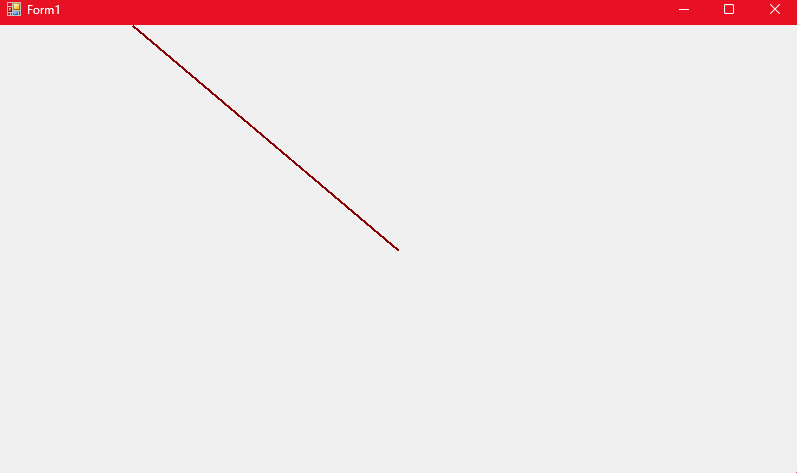


Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Бегущая строка.

Листинг программы:

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Task\_2

{

public partial class Form1 : Form

{

Graphics g;

Bitmap baner;

Rectangle rect;

public Form1()

{

InitializeComponent();

try

{

baner = new Bitmap(@"..\..\baner.png");

}

catch (Exception e)

{

MessageBox.Show("Ошибка загрузки файла баннера\n" + e.ToString(), "Баннер");

this.Close();

return;

}

g = this.CreateGraphics();

rect.X = 0;

rect.Y = 0;

rect.Width = baner.Width;

rect.Height = baner.Height;

timer1.Interval = 50;

timer1.Enabled = true;

}

private void timer1\_Tick(object sender, EventArgs e)

{

ClearForm();

rect.X += 1;

if (Math.Abs(rect.X) > rect.Width)

{

rect.X += rect.Width;

}

for (int i = 0; i <= Convert.ToInt16(this.ClientSize.Width / rect.Width) + 1; i++)

{

g.DrawImage(baner, rect.X + i \* rect.Width, rect.Y);

}

if (rect.X == this.Width)

rect.X = 0;

}

private void ClearForm()

{

g.FillRectangle(new SolidBrush(this.BackColor), new Rectangle(0, 0, this.Width, this.Height));

}

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

{

if ((e.Y < rect.Y + rect.Height) && (e.Y > rect.Y))

{

if (timer1.Enabled != false)

{

timer1.Enabled = false;

}

}

else

{

if (timer1.Enabled != true)

{

timer1.Enabled = true;

}

}

}

}

}

Таблица 2.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

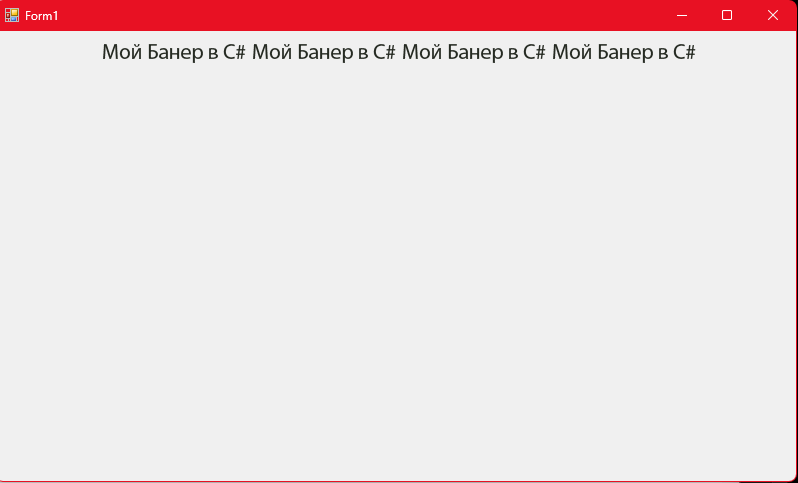


Рисунок 2.1 – Результат работы программы

Источник: собственная разработка

Задание 3. Движение по траектории

Листинг программы:

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

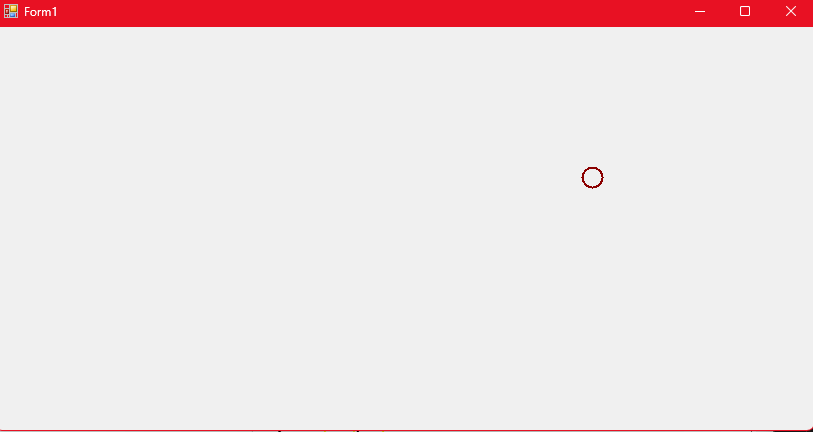


Рисунок 3.1 – Результат работы программы

Источник: собственная разработка

Задание 4. Полет.

Листинг программы:

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Task\_4

{

public partial class Form1 : Form

{

Bitmap sky, plane;

Graphics g;

int dx;

Rectangle rct;

Boolean demo = true;

Random rnd;

public Form1()

{

InitializeComponent();

try

{

sky = new Bitmap(@"..\..\sky.bmp"); // небо

plane = new Bitmap(@"..\..\plane.bmp"); // самолет

this.BackgroundImage = new Bitmap(@"..\..\sky.bmp");

}

catch (Exception exception)

{

MessageBox.Show(exception.ToString(), "Полет", MessageBoxButtons.OK, MessageBoxIcon.Error);

this.Close();

return;

}

plane.MakeTransparent();

this.ClientSize = new Size(new Point(BackgroundImage.Width, BackgroundImage.Height));

this.FormBorderStyle = FormBorderStyle.FixedSingle;

this.MaximizeBox = false;

g = Graphics.FromImage(BackgroundImage);

rnd = new Random();

rct.X = -40;

rct.Y = 20 + rnd.Next(20);

rct.Width = plane.Width;

rct.Height = plane.Height;

dx = 2;

timer1.Interval = 20;

timer1.Enabled = true;

}

private void timer1\_Tick(object sender, EventArgs e)

{

g.DrawImage(sky, new Point(0, 0));

if (rct.X < this.ClientRectangle.Width)

rct.X += dx;

else

{

rct.X = -40;

rct.Y = 20 +

rnd.Next(this.ClientSize.Height - 40 - plane.Height);

dx = 2 + rnd.Next(4);

}

g.DrawImage(plane, rct.X, rct.Y);

if (!demo)

this.Invalidate(rct);

else

{

Rectangle reg = new Rectangle(20, 20, sky.Width - 40, sky.Height - 40);

g.DrawRectangle(Pens.Black,

reg.X, reg.Y, reg.Width - 1, reg.Height - 1);

this.Invalidate(reg);

}

}

}

}

Таблица 4.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

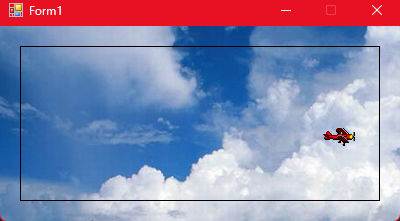


Рисунок 4.1 – Результат работы программы

Источник: собственная разработка

Задание 5. Разработайте программу анимации взлета ракеты. Старт осуществляется по нажатию специальной «красной» кнопки.

Листинг программы:

using System;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Windows.Forms;

namespace Task\_5

{

public partial class Form1 : Form

{

int y = 530;

public Form1()

{

InitializeComponent();

}

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

{

Graphics g = e.Graphics;

Graphics g1 = e.Graphics;

Graphics g2 = e.Graphics;

Graphics g3 = e.Graphics;

g.Clear(Color.Blue);

SolidBrush peg = new SolidBrush(Color.Red);

SolidBrush peg1 = new SolidBrush(Color.Yellow);

SolidBrush peg2 = new SolidBrush(Color.Yellow);

SolidBrush peg3 = new SolidBrush(Color.Aqua);

GraphicsPath gp = new GraphicsPath(FillMode.Winding);

GraphicsPath gp1 = new GraphicsPath(FillMode.Winding);

GraphicsPath gp2 = new GraphicsPath(FillMode.Winding);

GraphicsPath gp3 = new GraphicsPath(FillMode.Winding);

gp.AddPolygon(new Point[] { new Point(450, y), new Point(465, y + 100), new Point(485, y + 100), new Point(500, y), new Point(475, y - 30), new Point(450, y) });

gp1.AddPolygon(new Point[] { new Point(465, y + 100), new Point(445, y + 120), new Point(445, y + 90), new Point(460, y + 70), new Point(465, y + 100) });

gp2.AddPolygon(new Point[] { new Point(485, y + 100), new Point(505, y + 120), new Point(505, y + 90), new Point(490, y + 70), new Point(485, y + 100) });

gp3.AddEllipse(465, y + 5, 20, 20);

g.FillPath(peg, gp);

g1.FillPath(peg1, gp1);

g2.FillPath(peg2, gp2);

g3.FillPath(peg3, gp3);

}

private void button\_start\_Click(object sender, EventArgs e)

{

timer1.Start();

}

int H = 0;

private void timer1\_Tick(object sender, EventArgs e)

{

y--;

H++;

if (H == 680)

{

timer1.Stop();

H = 0;

y = 530;

}

Invalidate();

}

}

}

Таблица 5.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

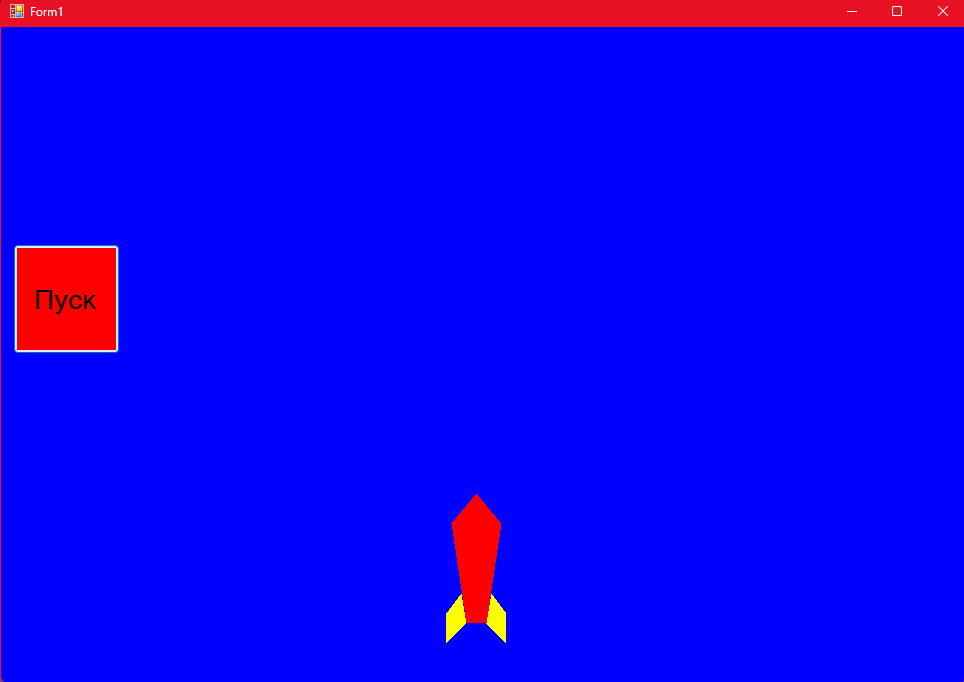


Рисунок 5.1 – Результат работы программы

Источник: собственная разработка