

Подготовка к лабораторной работе

1. Дескриптор контекста устройства и функции для его получения:

a. **e.Graphics**

экземпляр класса `Graphics`, который представляет собой поверхность рисования. Именно на ней и нужно рисовать

b. **this.CreateGraphics()**

этот метод возвращает объект рисования для формы или элемента управления, для которого он был вызван. В данном случае он вызывается для `this`, а значит, вернет объект рисования текущей формы

c. **Graphics.FromHwnd()**

статичный метод для создания объекта `Graphics` на основе `Hwnd` значения компонента

2. Выбор и создание кисти:

a. **System.Drawing.Brushes**

кисти для каждого из стандартных цветов

b. **SolidBrush**

кисть означает, что область должна быть заполнена сплошным цветом

c. **System.Drawing.Drawing2D.HatchBrush**

штриховая кисть, которая заполняет область, рисуя некоторый шаблон

d. **System.Drawing.TextureBrush**

кисть, которая использует картинку, чтобы заполнить содержимое фигуры

e. **LinearGradientBrush**

кисть заполняет область цветом, плавно изменяемым по экрану

3. Выбор и создание карандаша.

Стандартные карандаши: `System.Drawing.Pens`

Создание карандаша. Например: `p = new Pen (Color.Aquamarine, 3);`

Ширина: `p.Width = 5;`

Цвет: `p.Color = Color.Coral;`

Стиль: `p.DashStyle = System.Drawing.Drawing2D.DashStyle.DashDot;`

4. Функции рисования.

`DrawArc()` — нарисовать дугу;

`DrawLine()` — нарисовать линию;

`DrawEllipse()` — нарисовать эллипс;

`DrawRectangle()` — нарисовать прямоугольник;

`DrawPolygon()` — нарисовать многоугольник по массиву точек;

`DrawImage()` — нарисовать картинку;

`DrawString()` — отобразить строку текста;

`FillEllipse()`, `FillRectangle()`, `FillPolygon()`, `FillRegion()` — залить цветом эллипс, многоугольник, прямоугольник или область.

5. Прямоугольники, регионы, область рисования.

```
Rectangle r = new Rectangle (100,100,50,150);
```

```
Region reg = new Region(r);
```

Свойство **Clip** контекста устройства

регион (прямоугольная область), который определяет область рисования

6. Область отсечения и обновление окна: методы **Button.Refresh()** и **Invalidate()**;
Метод `Invalidate()` заставляет форму вызвать событие `Paint`.

7. Цвет (три и четыре параметра).

```
Color col=Color.FromArgb(255,255,100);
```

```
Color col=Color.FromArgb(100,255,255,100);
```

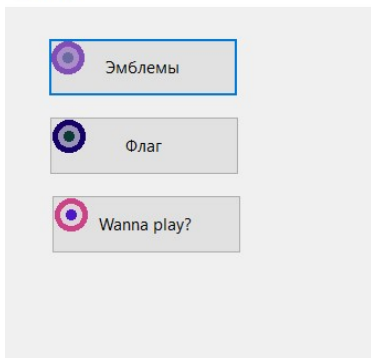
```
Color.FromArgb(прозрачность, красный, зеленый, синий);
```

Метод `Clear` – обновление области рисования и заливка заданным цветом

Ход работы

1. Придумать маленькую эмблему своей программы и нарисовать ее на всех кнопках. Картинка не должна исчезать после сворачивания или перекрытия окна.
Эмблемы будут рисоваться по событию `Paint` у кнопок.

Form1



```
public bool emb_paint = false; // флаги для включения / отключения отображения эмблем
public bool fl_paint = false;
public bool game_paint = false;

private void button1_Click(object sender, EventArgs e) // управляющая отображением кнопка
{
    emb_paint = !emb_paint;
    fl_paint = !fl_paint;
    game_paint = !game_paint;
}

private void DrawEmb(PaintEventArgs e) // функция отрисовки эмблем
{
    //
    //Graphics gr = Graphics.FromHwnd(butt.Handle);
    Graphics gr = e.Graphics;
    Random rnd = new Random(); // цвета элементов эмблемы будут рандомны
    int A = rnd.Next(0, 255);
    int R = rnd.Next(0, 255);
    int G = rnd.Next(0, 255);
    int B = rnd.Next(0, 255);
    Pen p = new Pen(Color.FromArgb(R, G, B), 5);
    gr.DrawEllipse(p, 5, 5, 25, 25);
    gr.DrawEllipse(new Pen(Color.FromArgb(G, A, R), 5), 15, 15, 5, 5);
    Brush br = new SolidBrush(Color.FromArgb(A, R, G, B));
    gr.FillEllipse(br, 5, 5, 25, 25);
}

private void button1_Paint(object sender, PaintEventArgs e)
{
    if (emb_paint) DrawEmb(e);
}

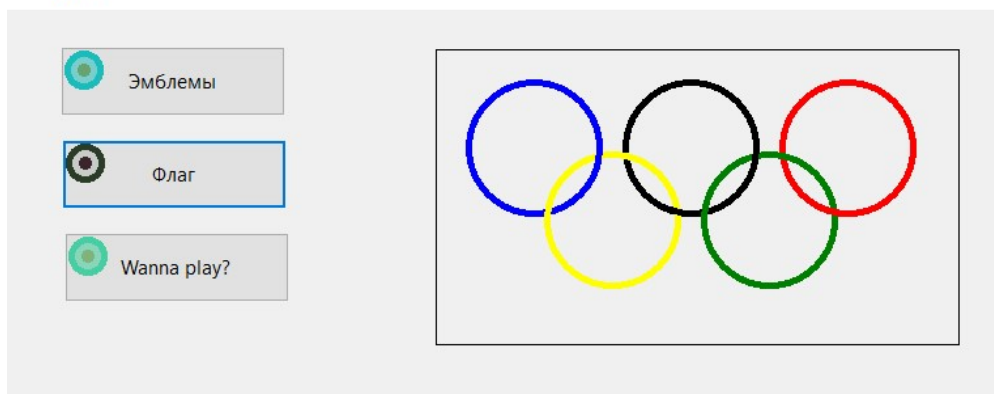
private void button2_Paint(object sender, PaintEventArgs e)
{
    if (fl_paint) DrawEmb(e);
}

private void button3_Paint(object sender, PaintEventArgs e)
{
    if (game_paint) DrawEmb(e);
}
```

2. Нарисовать флаг олимпийских игр.

По событию Paint формы.

Form1



```
public bool fl = false;

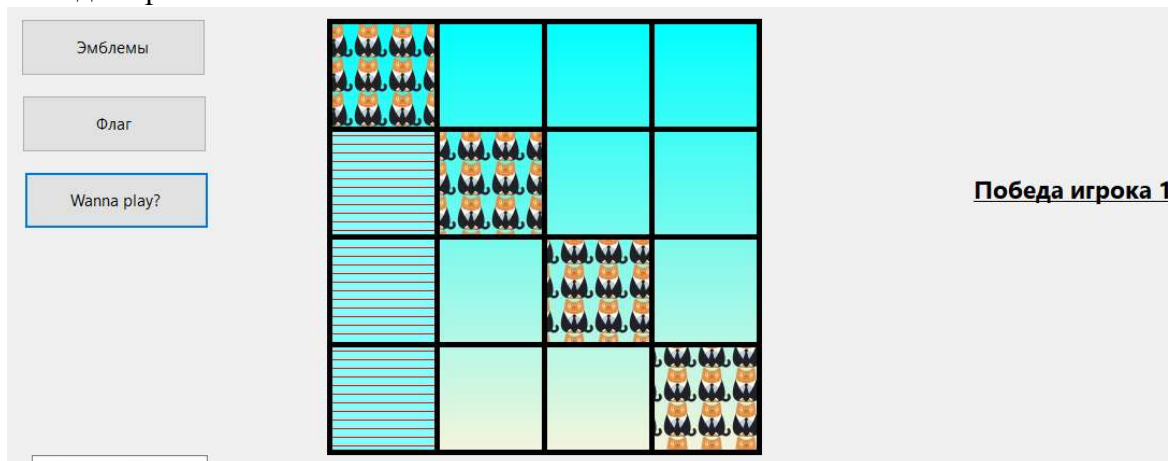
private void button2_Click(object sender, EventArgs e)
{
    //
    fl = !fl;
    Refresh();
}

private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics gr = e.Graphics;
    if (fl)
    {
        gr.DrawRectangle(new Pen(Color.Black, 1), new Rectangle(330, 30, 400, 225));
        gr.DrawEllipse(new Pen(Color.Blue, 5), new Rectangle(355, 55, 100, 100));
        gr.DrawEllipse(new Pen(Color.Black, 5), new Rectangle(475, 55, 100, 100));
        gr.DrawEllipse(new Pen(Color.Red, 5), new Rectangle(595, 55, 100, 100));
        gr.DrawEllipse(new Pen(Color.Yellow, 5), new Rectangle(415, 110, 100, 100));
        gr.DrawEllipse(new Pen(Color.Green, 5), new Rectangle(535, 110, 100, 100));
        gr.DrawLine(new Pen(Color.Blue, 5), new Point(454, 115), new Point(455, 105));
        gr.DrawLine(new Pen(Color.Black, 5), new Point(509, 152), new Point(520, 155));
        gr.DrawLine(new Pen(Color.Black, 5), new Point(574, 115), new Point(575, 104));
        gr.DrawLine(new Pen(Color.Red, 5), new Point(629, 152), new Point(639, 155));
    }
}
```

3. Создать программу «Игра», похожую на крестики-нолики. Игровое поле – клетки-прямоугольники. При щелчке на клетке визуализируется ход игрока номер 1, следующий щелчок совершает игрок номер 2. Выигрывает тот, кто выбрал 4 клетки подряд. При создании интерфейса для игры использовать текстурные и градиентные кисти, а также режим прозрачности.

Фон для поля игры – градиентная кисть, ход 1-ого игрока – текстурная кисть, ход 2-ого игрока – штриховая кисть.

Победа игрока 1:



Эмблемы

Флаг

Wanna play?

Победа игрока 1

Победа игрока 2:

Эмблемы

Флаг

Wanna play?

Победа игрока 2

Эмблемы

Флаг

Wanna play?

Победа игрока 2

Ничья:

Эмблемы

Флаг

Wanna play?

Ничья!

```

public bool game = false; // отрисовка поля до первого хода
public bool start = false; // останавливаем игру и делаем поле недоступным, как только кто-то победил или
ничья
public int player; // номер игрока, делающего ход
public int steps; // счётчик ходов в целом

    public Rectangle[,] rects = new Rectangle[4, 4] // прямоугольники игры для отрисовки ходов
    { { new Rectangle(330, 30, 100, 100), new Rectangle(430, 30, 100, 100), new Rectangle(530, 30,
100, 100), new Rectangle(630, 30, 100, 100) },
      {new Rectangle(330, 130, 100, 100), new Rectangle(430, 130, 100, 100), new Rectangle(530,
130, 100, 100), new Rectangle(630, 130, 100, 100) },
      {new Rectangle(330, 230, 100, 100), new Rectangle(430, 230, 100, 100), new Rectangle(530,
230, 100, 100), new Rectangle(630, 230, 100, 100) },
      {new Rectangle(330, 330, 100, 100), new Rectangle(430, 330, 100, 100), new Rectangle(530,
330, 100, 100), new Rectangle(630, 330, 100, 100) } };

    public int[,] moves = new int[4, 4]; // матрица игры

private void button3_Click(object sender, EventArgs e)
{
    //
    game = !game;
    start = true;
    Array.Clear(moves, 0, 16); // обнуляем игру
    player = 1;
    steps = 0;
    lb1.Text = "Ходит игрок 1";
    Refresh();
}

private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics gr = e.Graphics;
    if (game && !fl) // если флаг не нарисован
    { // фон для поля игры - градиентная кисть
        LinearGradientBrush br_grad = new LinearGradientBrush(new Rectangle(330, 30, 400, 400),
            Color.Aqua, Color.Beige, LinearGradientMode.Vertical);
        gr.FillRectangle(br_grad, 330, 30, 400, 400);
        gr.DrawRectangle(new Pen(Color.Black, 5), new Rectangle(330, 30, 400, 400));
        gr.DrawLine(new Pen(Color.Black, 5), new Point(430, 30), new Point(430, 430));
        gr.DrawLine(new Pen(Color.Black, 5), new Point(530, 30), new Point(530, 430));
        gr.DrawLine(new Pen(Color.Black, 5), new Point(630, 30), new Point(630, 430));
        gr.DrawLine(new Pen(Color.Black, 5), new Point(330, 130), new Point(730, 130));
        gr.DrawLine(new Pen(Color.Black, 5), new Point(330, 230), new Point(730, 230));
        gr.DrawLine(new Pen(Color.Black, 5), new Point(330, 330), new Point(730, 330));
    }
}

private void player_1(Rectangle rect) // отрисовка хода игрока 1 - текстурная кисть
{
    Bitmap image1 = (Bitmap) Image.FromFile(@"C:\Users\" +
@"user\source\repos\lab_6_sp\picture.png",
        true);
    TextureBrush texture = new TextureBrush(image1);
    Graphics gr = this.CreateGraphics();
    gr.FillRectangle(texture, rect);
    gr.DrawRectangle(new Pen(Color.Black, 5), rect);
    gr.Dispose();
}

private void player_2(Rectangle rect) // отрисовка хода игрока 2 - штриховая кисть
{
    HatchBrush hb = new HatchBrush(HatchStyle.Horizontal,
        Color.Red,
        Color.FromArgb(255, 128, 255, 255));
    Graphics gr = this.CreateGraphics();
    gr.FillRectangle(hb, rect);
    gr.DrawRectangle(new Pen(Color.Black, 5), rect);
    gr.Dispose();
}

```

```

private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    if (start)
    {
        //
        if (e.X >= 330 && e.X < 730 && e.Y >= 30 && e.Y < 430)
        {
            int i = (e.X - 330) / 100;
            int j = (e.Y - 30) / 100;
            if (moves[j, i] == 0)
            {
                if (player == 1)
                {
                    player_1(rects[j, i]);
                    moves[j, i] = 1;
                }
                else
                {
                    player_2(rects[j, i]);
                    moves[j, i] = 2;
                }

                steps++; // количество общих ходов увеличивается
                if (steps == 16) // все клетки заполнены - ничья
                {
                    lb1.Text = "Ничья!";
                    start = false;
                    return;
                }

                for (int k = 0; k < 4; k++) // горизонтальная или вертикальная линия заполнена -
                один из игроков победил
                {
                    if (moves[0, k] == moves[1, k] && moves[1, k] == moves[2, k]
                        && moves[2, k] == moves[3, k] && moves[0, k] != 0)
                    {
                        lb1.Text = "Победа игрока " + moves[0, k];
                        start = false;
                        return;
                    }

                    if (moves[k, 0] == moves[k, 1] && moves[k, 1] == moves[k, 2]
                        && moves[k, 2] == moves[k, 3] && moves[k, 0] != 0)
                    {
                        lb1.Text = "Победа игрока " + moves[k, 0];
                        start = false;
                        return;
                    }
                }

                if (moves[0, 0] == moves[1, 1] && moves[1, 1] == moves[2, 2] &&
                    moves[2, 2] == moves[3, 3] && moves[0, 0] != 0) // главная диагональ
                {
                    lb1.Text = "Победа игрока " + moves[0, 0];
                    start = false;
                    return;
                }

                if (moves[0, 3] == moves[1, 2] && moves[1, 2] == moves[2, 3] &&
                    moves[2, 3] == moves[3, 0] && moves[0, 3] != 0) // побочная диагональ
                {
                    lb1.Text = "Победа игрока " + moves[0, 3];
                    start = false;
                    return;
                }

                if (player == 1) // меняем игрока
                    player = 2;
                else
                    player = 1;

                lb1.Text = "Ходит игрок " + player;
            }
        }
    }
}

```

}
}
}
}