Лабораторна робота №2

Тема: Двовимірне векторне зображення

Мета: Написати програму виведення на екран векторних малюнків з використанням перетворень переносу, масштабування та повороту.

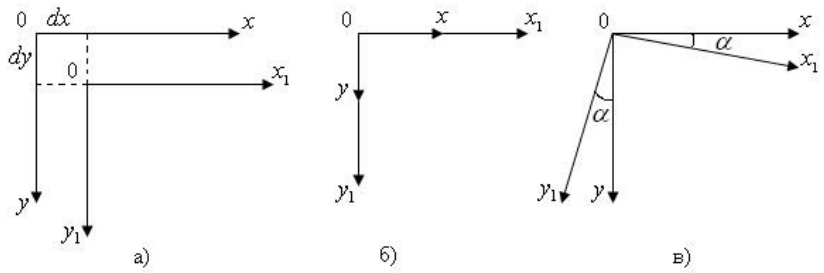
Теоретичні відомості

У комп'ютерній графіці перетворення координат застосовують для зміни масштабу зображення, отримання симетричних частин графічних об'єктів, дублювання частин зображення тощо. Перетворення координат об'єкта — це їх переоб-числення за певними формулами. У подальшому розглядатимемо лише лінійні перетворення координат на площині, під час яких зберігаються такі геометричні властивості об'єктів: прямі лінії залишаються прямими, зберігається паралельність прямих, а також відношення площ геометричних фігур. Основними лінійними перетвореннями є паралельне перенесення, розтягування (стискання) і поворот геометричних об'єктів.

Припустимо, що у декартовій системі координат задана плоска геометрична фігура, (x,y) — координати однієї з її точок. Нехай (x1,y1) - координати тієї самої точки фігури після їх перетворення. Наведемо формули паралельного перенесення геометричної фігури на dx одиниць уздовж осі X і на dy одиниць уздовж осі Y (формула а), розтягування уздовж осі X у kx разів і вздовж осі У у ky разів (формула 6) і повороту навколо початку координат на кут α (формула в).

Лінійні перетворення проілюстровано на рис. 5.3. Необхідно зазначити, що будь-яке перетворення координат об'єкта можна розглядати як перетворення координатних осей.

а) $\begin{cases} x_1 = x \pm dx; \\ y_1 = y \pm dy; \end{cases}$ б) $\begin{cases} x_1 = x \cdot k_x; \\ y_1 = y \cdot k_y; \end{cases}$ в) $\begin{cases} x_1 = x \cdot \cos\alpha + y \cdot \sin\alpha; \\ x_1 = -x \cdot \sin\alpha + y \cdot \cos\alpha. \end{cases}$



а) б) в)

Хід роботи

1. Створіть та збережіть новий проект в середовищі Lazarus.

2. Додайте на головне вікно компонент TImage та TButton.

3. Подвійним кліком на кнопці додайте процедуру Form1.onButton1Click.

4. Створіть довільний простий малюнок (хатка, кицька, пацюк, тощо). Малюнок повинен бути унікальним для кожного виконавця роботи.

5. Виведіть малюнок в різному масштабуванні та кутах повороту.

6. За допомогою інших елементів керування зробіть налаштування що до поточного масштабу та куту повороту з метою змінювання одного з зображень без перекомпіляції програми.

7. Створіть композицію з намальованих елементів, щоб малюнок виглядав цілісним.

8. Результат роздрукуйте та додайте до звіту разом з текстом програми.

9. Дайте відповіді на контрольні питання.

10. В разі виконання роботи на поточній парі дозволяється використання електронного звіту з усними відповідями на контрольні питання.

11. Зробіть висновки що до досяжності мети поставленої в лабораторній роботі.

Контрольні питання:

1. Поясніть спотворення еліпсу яке зображає око песика з теоретичних відомостей.

2. Що ви запропонуєте для використання почергово декількох перетворювачів координат?

3. Що потрібно для малювання графічного примітиву “прямокутник” з використанням перетворення обертання?

4. Які зміни потрібно зробити, щоб обертання відбувалося в зворотньому напрямку?

5. Який максимальний кут обертання зображення?

6. Що ви зробите, щоб намалювати зображення догори ногами без використання обертання?

Додаток А. Песики.

```
unit Unit1;
{$mode objfpc}{$H+}
interface
uses
    Classes, SysUtils, FileUtil, Forms, Controls, Graphics,
Dialogs, ExtCtrls,   StdCtrls;

type
  { TForm1 }
  TTransformer = class
    public
      x, y: Integer;
      procedure SetXY(dx,dy: Integer);
  end;

  TForm1 = class(TForm)
    Button1: TButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
  private
    { private declarations }
  public
    { public declarations }
    Tra: TTransformer;
    procedure MLine(x1,y1,x2,y2:Integer);
    procedure MEllipse(x1,y1,x2,y2:Integer);
    procedure DrawPesik;
  end;

var
  Form1: TForm1;
```

```
implementation

{$R *.lfm}
{ TForm1 }
procedure TForm1.DrawPesik;
var i: Integer;
begin
  Image1.Canvas.Pen.Color:=clBlack;
  for i:=0 to 7 do begin Mline(i*2,10,i*2,20); end;
  for i:=0 to 7 do begin MLine(14+i*2,0,14+i*2,20); end;
  for i:=0 to 20 do begin MLine(24+i*2,20,24+i*2,40); end;
  for i:=0 to 8 do begin MLine(62+i*2,20,62+i*2,25); end;
  Image1.Canvas.Brush.Color:=clWhite;
  MEllipse(12,3,17,8);
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  Image1.Canvas.Brush.Color:=clWhite;
  Image1.Canvas.FillRect(0,0,Image1.Width,Image1.Height);
  Tra.SetXY(100,100);   DrawPesik;
  Tra.SetXY(150,150);   DrawPesik;
  Tra.SetXY(200,200);   DrawPesik;
  Tra.SetXY(250,100);   DrawPesik;
  Tra.SetXY(300,150);   DrawPesik;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin   Tra := TTransformer.Create; end;

procedure TForm1.FormDestroy(Sender: TObject);
begin   Tra.Destroy; end;

procedure TForm1.MLine(x1,y1,x2,y2:Integer);
begin
  Image1.Canvas.Line(x1+Tra.x,
```

```
                        y1+Tra.y,
                        x2+Tra.x,
                        y2+Tra.y);
end;

procedure TForm1.MEllipse(x1,y1,x2,y2:Integer);
begin
  Image1.Canvas.Ellipse(x1+Tra.x,
                          y1+Tra.y,
                          x2+Tra.x,
                          y2+Tra.y);
end;

procedure TTransformer.SetXY(dx,dy: Integer);
begin   x:=dx; y:=dy; end;

end.
```

Додаток Б. Масштабовані песики.

```
unit Unit1;

{$mode objfpc}{$H+}

interface

uses
    Classes, SysUtils, FileUtil, Forms, Controls, Graphics,
Dialogs, ExtCtrls,
  StdCtrls;

type

  { TForm1 }

  TTransformer = class
    public
      x, y: Integer;
      mx, my: Double;
      procedure SetXY(dx,dy: Integer);
      procedure SetMXY(masX, masY: Double);
      function GetX(oldX:Integer):Integer;
      function GetY(oldY:Integer):Integer;
  end;

  TForm1 = class(TForm)
    Button1: TButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
  private
    { private declarations }
  public
```

```
    { public declarations }
    Tra: TTransformer;
    procedure MLine(x1,y1,x2,y2:Integer);
    procedure MEllipse(x1,y1,x2,y2:Integer);
    procedure DrawPesik;
  end;

var
  Form1: TForm1;

implementation

{$R *.lfm}

{ TForm1 }

procedure TForm1.DrawPesik;
var i: Integer;
begin
  Image1.Canvas.Pen.Color:=clBlack;
  for i:=0 to 7 do begin
    MLine(i*2,10,i*2,20);
  end;
  for i:=0 to 7 do begin
    MLine(14+i*2,0,14+i*2,20);
  end;
  for i:=0 to 20 do begin
    MLine(24+i*2,20,24+i*2,40);
  end;
  for i:=0 to 8 do begin
    MLine(62+i*2,20,62+i*2,25);
  end;
  Image1.Canvas.Brush.Color:=clWhite;
  MEllipse(12,3,17,8);
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Image1.Canvas.Brush.Color:=clWhite;
  Image1.Canvas.FillRect(0,0,Image1.Width,Image1.Height);
  Tra.SetXY(50,100); Tra.SetMXY(1.0,1.0);
  DrawPesik;
  Tra.SetXY(100,150); Tra.SetMXY(1.4,1.4);
  DrawPesik;
  Tra.SetXY(150,220); Tra.SetMXY(1.8,1.8);
  DrawPesik;
  Tra.SetXY(200,100); Tra.SetMXY(0.8,0.8);
  DrawPesik;
  Tra.SetXY(250,150); Tra.SetMXY(1.4,1.4);
  DrawPesik;
  Tra.SetXY(300,300); Tra.SetMXY(-1.0,1.0);
  DrawPesik;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  Tra := TTransformer.Create;
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
  Tra.Destroy;
end;

procedure TForm1.MLine(x1,y1,x2,y2:Integer);
begin
  Image1.Canvas.Line(Tra.GetX(x1),
                     Tra.GetY(y1),
                     Tra.GetX(x2),
                     Tra.GetY(y2));
end;
```

```
procedure TForm1.MEllipse(x1,y1,x2,y2:Integer);
begin
  Image1.Canvas.Ellipse(Tra.GetX(x1),
                        Tra.GetY(y1),
                        Tra.GetX(x2),
                        Tra.GetY(y2));
end;

{ TTransformer }

procedure TTransformer.SetXY(dx,dy: Integer);
begin
  x:=dx; y:=dy;
end;

procedure TTransformer.SetMXY(masX, masY: Double);
begin
  mx:=masX; my:=masY;
end;

function TTransformer.GetX(oldX:Integer):Integer;
begin
  GetX := Round( mx*oldX + x);
end;

function TTransformer.GetY(oldY:Integer):Integer;
begin
  GetY := Round( my*oldY + y);
end;

end.
```

Додаток В. Собаки з поворотом

```
unit Unit1;

{$mode objfpc}{$H+}

interface

uses
    Classes, SysUtils, FileUtil, Forms, Controls, Graphics,
Dialogs, ExtCtrls,
  StdCtrls;

type

  { TForm1 }

  TTransformer = class
    public
      x, y: Integer;
      mx, my: Double;
      alpha: Double;
      procedure SetXY(dx,dy: Integer);
      procedure SetMXY(masX, masY: Double);
      procedure SetAlpha(a: Double);
      function GetX(oldX, oldY:Integer):Integer;
      function GetY(oldX, oldY:Integer):Integer;
  end;

  TForm1 = class(TForm)
    Button1: TButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
  private
```

```
    { private declarations }
  public
    { public declarations }
    Tra: TTransformer;
    procedure MLine(x1,y1,x2,y2:Integer);
    procedure MEllipse(x1,y1,x2,y2:Integer);
    procedure DrawPesik;
  end;

var
  Form1: TForm1;

implementation

{$R *.lfm}

{ TForm1 }

procedure TForm1.DrawPesik;
var i: Integer;
begin
  Image1.Canvas.Pen.Color:=clBlack;
  for i:=0 to 7 do begin
    MLine(i*2,10,i*2,20);
  end;
  for i:=0 to 7 do begin
    MLine(14+i*2,0,14+i*2,20);
  end;
  for i:=0 to 20 do begin
    MLine(24+i*2,20,24+i*2,40);
  end;
  for i:=0 to 8 do begin
    MLine(62+i*2,20,62+i*2,25);
  end;
  Image1.Canvas.Brush.Color:=clWhite;
  MEllipse(12,3,17,8);
```

```
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  Image1.Canvas.Brush.Color:=clWhite;
  Image1.Canvas.FillRect(0,0,Image1.Width,Image1.Height);
  Tra.SetAlpha(0.0);
  Tra.SetXY(50,100); Tra.SetMXY(1.0,1.0);
  DrawPesik;
  Tra.SetXY(100,150); Tra.SetMXY(1.4,1.4);
  DrawPesik;
  Tra.SetXY(150,220); Tra.SetMXY(1.8,1.8);
  DrawPesik;
  Tra.SetXY(200,100); Tra.SetMXY(0.8,0.8);
  DrawPesik;
  Tra.SetAlpha(PI/4.0); //Тут змінено кут
  Tra.SetXY(250,150); Tra.SetMXY(1.4,1.4);
  DrawPesik;
  Tra.SetXY(300,350); Tra.SetMXY(-1.0,1.0);
  DrawPesik;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  Tra := TTransformer.Create;
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
  Tra.Destroy;
end;

procedure TForm1.MLine(x1,y1,x2,y2:Integer);
begin
  Image1.Canvas.Line(Tra.GetX(x1,y1),
                     Tra.GetY(x1,y1),
```

```
                    Tra.GetX(x2,y2),
                    Tra.GetY(x1,y2));
end;

procedure TForm1.MEllipse(x1,y1,x2,y2:Integer);
begin
  Image1.Canvas.Ellipse(Tra.GetX(x1,y1),
                    Tra.GetY(x1,y1),
                    Tra.GetX(x2,y2),
                    Tra.GetY(x2,y2));
end;

{ TTransformer }
procedure TTransformer.SetXY(dx,dy: Integer);
begin   x:=dx; y:=dy; end;

procedure TTransformer.SetMXY(masX, masY: Double);
begin   mx:=masX; my:=masY; end;

function TTransformer.GetX(oldX, oldY:Integer):Integer;
begin
  GetX := Round( mx*oldX*cos(alpha) - my*oldY*sin(alpha) + x);
end;

function TTransformer.GetY(oldX, oldY:Integer):Integer;
begin
  GetY := Round( mx*oldX*sin(alpha) + my*oldY*cos(alpha) + y);
end;

procedure TTransformer.SetAlpha(a: Double);
begin
  alpha:=a;
end;

end.
```