

Rīgas Tehniskā Universitāte  
Datorzinātnes un Informācijas Tehnoloģijas fakultāte  
Lietišķo datorsistēmu institūts

**„Datorgrafikas un attēlu apstrādes pamati”  
Kontroldarbs Nr.1  
„Bresenhema līniju algoritms”**

D.I.T.F  
RDB 1.kurss 14.grupa  
Sergejs Terentjevs  
Studenta apl. 061RDB140

2006./07 mācību gads

## 1.Darba uzdevums

Izmantojot Bresenhema algoritmu līniju attēlošanai, izstrādāt programmu, kas uzdotiem sākuma un gala punktiem konstruētu līniju divdimensiju rastrā.

## 2.Teorētiskais pamatojums

Pieņemsim, ka mums ir jāattēlo līnija, kura konstruēsies divdimensiju rastrā:

Pirmkārt, ir nepieciešams atrast līnijas kopējo garumu, kuru mēs iegūsim no līnijas beigu punktiem atņemot līnijas sākumpunktus

**$dx:=abs(xe-xb);$**

**$dy:=abs(ye-yb),$**

kur  **$dx, dy$**  – delta no x un y,

**$xb, yb$**  – taisnes sākumpunkti,

**$xe, ye$**  – taisnes beigu punkti,

konstante  **$abs$**  – norāda, ka rezultātam jābūt pozitīvam.

Turpmāk mums ir jānosaka līnijas jeb taisnes virziens:

**$if\ xe - xb < 0\ then\ ix := -1$**

**$else\ ix:=1;$**

**$if\ ye - yb < 0\ then\ iy := -1$**

**$else\ iy := 1,$**

kur  **$ix, iy$**  – taisnes virzieni (ja 1, tad līnija konstruēsies pa labi, ja -1, tad līnija konstruēsies pa kreisi)

Tad mums jānosaka, kura rastrā tiks attēlota līnija

**$dx>=dy,$**

zīmes pamatojoties uz x asi.

**$dy>=dx,$**

zīmes pamatojoties uz y asi.

Tādējādi mēs iegūsim līniju, kurā konstruējas abos rastros.

Vēl mums ir jānosaka sākotnējais risinājušais parametrs un pikseli, kuriem degot, veidosies taisne un taisnes virziens:

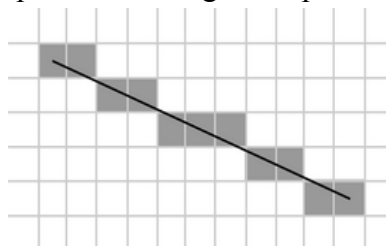
Sākotnējo risinājušo parametru nosakām pēc formulas

**$p:=2*dy-dx$**

vai otrajā rastrā

**$p:=2*dx-dy,$**

Tad jā  $P > 0$ , tad līnija konstruēsies pa labi uz augšu vai pa labi un taisni, bet jā virziens būs negatīvs, līnija tiks konstruēta pa kreisi uz augšu vai pa kreisi taisni.



Otrajā rastrā līnija konstruēties balstoties uz y vērtībām (risinājušais parametrs projektēties uz y ass).

### 3. Programmas Kods

```
unit taisne;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls;
```

```
type
```

```
  TForm1 = class(TForm)  
    Edit1: TEdit;  
    Edit2: TEdit;  
    Edit3: TEdit;  
    Edit4: TEdit;  
    Label1: TLabel;  
    Label2: TLabel;  
    Label3: TLabel;  
    Label4: TLabel;  
    Button1: TButton;  
    procedure Button1Click(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;
```

```
var
```

```
  Form1: TForm1;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure Brezen(xb,yb,x,ye:integer);
```

```
var x,y,dx,dy,ix,iy,p,i:integer;
```

```
begin
```

```
  dx:=abs(xe-xb);
```

```
  dy:=abs(ye-yb);
```

```
  x:=xb;
```

```
  y:=yb;
```

```
  if xe-xb<0 then ix:=-1
```

```
  else ix:=1;
```

```
  if ye-yb<0 then iy:=-1
```

```

else iy:=1;

if dx>=dy then
begin

p:=2*dy-dx;
for i:=1 to dx do
begin

if p>0 then
begin
x:=x+ix;
y:=y+iy;
p:=p+2*(dy-dx);
end
else
begin
x:=x+ix;
y:=y;
p:=p+2*dy;
end;
Form1.Canvas.Pixels[x,y]:=$000000;
end;
end;

if dy>=dx then
begin

p:=2*dx-dy;
for i:=1 to dy do
begin

if p>0 then
begin
y:=y+iy;
x:=x+ix;
p:=p+2*(dx-dy);
end
else
begin
y:=y+iy;
x:=x;
p:=p+2*dx;
end;
Form1.Canvas.Pixels[x,y]:=$000000;
end;
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
var x1,x2,y1,y2:integer;
begin
x1:=strtoint(Edit1.text);
y1:=strtoint(Edit2.text);

```

```
x2:=strtoint(Edit3.text);  
y2:=strtoint(Edit4.text);  
  
Brezen(x1,y1,x2,y2);
```

```
end;
```

```
end.
```

#### **4. Secinājumi**

Ir iegūta strādājoša programma, kura konstruē līniju divdimensiju rastra.