

Rīgas Tehniskā Universitāte
Datorzinātnes un Informācijas Tehnoloģijas fakultāte
Lietišķo datorsistēmu institūts

**„Datorgrafikas un attēlu apstrādes pamati”
Kontroldarbs Nr.3
„Elipses līnijas veidošanas algoritms”**

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

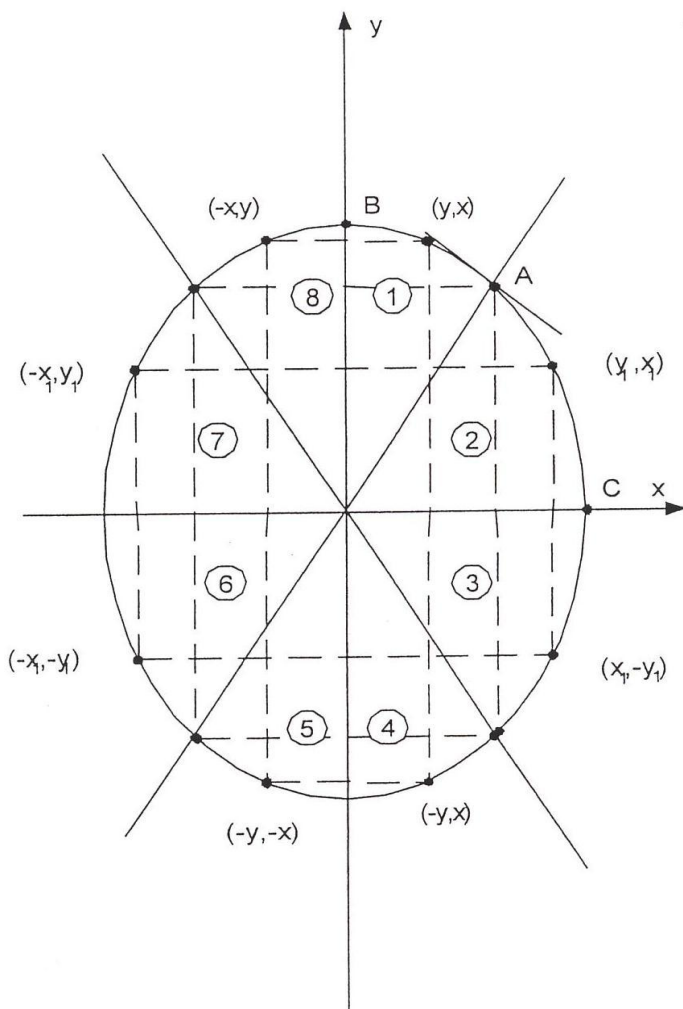
2006./07 mācību gads

1.Darba uzdevums

Izmantojot elipses līnijas veidošanas algoritmu, izstrādāt programmu, kura konstruē elipsi visas oktantās.

2.Teorētiskais pamatojums

Elipses zīmēšanas principi:



Lai attēlotu elipsi, sadalīsim elipsi astoņas simetrijas, pieņemsim, ka mums jāattēlo pirmais kvadrants:

Uzskatīsim, ka $r_x < r_y$ un sadalīsim pirmo kvadrantu divas daļas, sāksim veidot elipsi no sākumpunkta B līdz punktam A, kura pieskares slīpums būs vienāds ar -1. Tādējādi slīpumam paliekot mazākam par -1 veidosim uz y virzienu līdz punktam C

Turpmāk no 1 daļas var atrast simetriskus punktus daļai 4,5,8, no 2 daļas punktus 3,6,7 ($r_x > r_y$).

Elipses veidošanas algoritma pamatojums:

1. jānosaka rādītājus x un y un centru ($x_c=0, y_c=0$), tādējādi sākotnēja punkta koordinātes būs (0, r_y).
2. Definējam slīpumus ($Rx2y:=2*R_x2*y$, $Ry2x:=2*R_y2*x$) un rēķinām sākotnējo vērtību risinošam parametram pirmā daļā ($p:=Ry2-Rx2+0.25*R_x2$).
3. Katram x_k ($k=0$), pārbaudīsim:
 - ❖ Jā $p_{k+1} < 0$, tad nākamais pikselis būs (x_{k+1}, y_k) un $p_{k+1}:=p+Ry2x+Ry2$;
 - ❖ Citādi nākamais pikselis būs (x_{k+1}, y_k-1) un $p_{k+1}:=p+Ry2x-Rx2y+Ry2$.

4. Jā pieskares slīpums ir -1, tad aprēķinām sākotnējo vērtību risinājušam parametram otra daļā $p_{20} := Ry^2 * \text{sqr}(x+0.5) + Rx^2 * \text{sqr}(y-1) - Rx^2 * Ry^2$, kur x, y – gala punkts pirmā daļā.
5. Katra y_k ($k=0$), pārbaudīsim:
 - ❖ Jā $p_{2k} > 0$, tad nākamais pikselis būs $(x_k, y_k - 1)$ un $p_{2k+1} := p - Rx^2y + Rx^2$;
 - ❖ Citādi nākamais pikselis būs $(x_k + 1, y_k - 1)$ un $p_{2k+1} := p + Ry^2x - Rx^2y + Rx^2$;
6. Meklējam simetriskos punktus citos kvadrantos.

3. Programmas kods

```
procedure Zimet(x,y,xc,yc:Integer);
begin
  Form1.Canvas.Pixels[x+xc,y+yc]:=$000000;
  Form1.Canvas.Pixels[-x+xc,y+yc]:=$000000;
  Form1.Canvas.Pixels[x+xc,-y+yc]:=$000000;
  Form1.Canvas.Pixels[-x+xc,-y+yc]:=$000000;
end;
```

```
procedure Elipse (xc,yc,rx,ry:Integer);
var x,y:Integer;
    Rx2y,Ry2x,Rx2,Ry2,p:Real;
begin
  x:=-1;
  y:=Ry;
  Rx2:=sqr(Rx);
  Ry2:=sqr(Ry);
  Rx2y:=2*Rx2*y;
  Ry2x:=2*Ry2*x;
  p:=Ry2-Rx2+0.25*Rx2;
  While Ry2x<=Rx2y do
    begin
      if p<0 then
        begin
          x:=x+1;
          Ry2x:=2*Ry2*x;
          p:=p+Ry2x+Ry2;
        end
      else
        begin
          x:=x+1;
          y:=y-1;
          Rx2y:=2*Rx2*y;
          Ry2x:=2*Ry2*x;
          p:=p+Ry2x-Rx2y+Ry2;
        end;
      Zimet(x,y,xc,yc);
    end;
  while y>=0 do
    begin
      If p>0 then
        begin
          y:=y-1;
```

```

        Rx2y:=2*Rx2*y;
        p:=p-Rx2y+Rx2;
    end
    else
    begin
        x:=x+1;
        y:=y-1;
        rx2y:=2*Rx2*y;
        ry2x:=2*Ry2*x;
        p:=p+Ry2x-Rx2y+Rx2;
    end;
    Zimet(x,y,xc,yc);

end;
end;

```

4. secinājumi

Pielietojot elipses līnijas veidošanas algoritmu ir iegūta strādājoša programma, kurā zīme elipsi visās astoņas simetrijas.