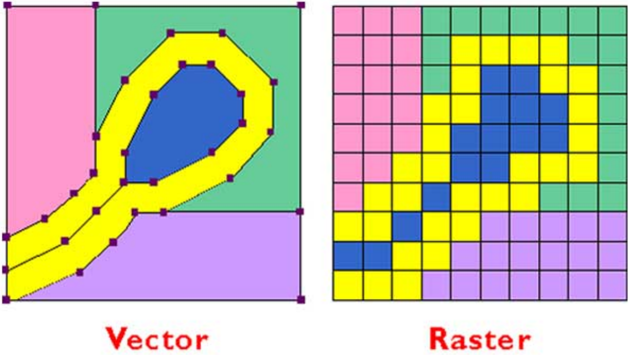
Računarska grafika
Algoritmi za crtanje 2D primitiva

Vesna Marinković

Osnovni koncepti sa prethodnog časa

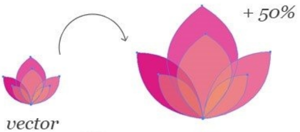
- *Računarska grafika* se bavi pravljenjem modela objekata na sceni i modela osvetljenja scene i na osnovu toga pravljenjem određenog pogleda na scenu
- Dve osnovne poddiscipline računarske grafike:
  - *Modelovanje* – izrada matematičke specifikacije objekata i njihovih vizuelnih svojstava, zadavanje modela svetlosti i pozicioniranje na sceni; hijerarhijski je zasnovano
  - *Renderovanje* – na osnovu modela objekata i modela ponašanja svetlosti pravi se realistična 2D slika; razlikujemo renderovanje unapred i renderovanje unazad
- Grafika može biti zasnovana na uzorku ili na geometriji
- Grafika može biti interaktivna i neinteraktivna

Vektorski i rasterski sistemi



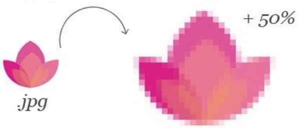
Vektorska grafika (eps, svg, ai, . . .)

- Slika je predstavljena kao kolekcija geometrijskih figura (tačke, prave, krive, poligoni,. . .)
- Za svaku figuru čuvaju se njeni parametri i njen položaj na slici
- Memorija koju slika zauzima zavisi od njenog sadržaja, često je manja od rasterske slike
- Skaliranje vektorske slike je jednostavno i sa savršenom preciznošću
- Vektorska grafika se koristi za fontove, logotipove, štampani materijal velikog formata, animacije, . . .
- Nije pogodna za predstavljanje fotografija ili fotorealističnih slika; potreban je poseban softver za njihovu obradu
- SVG tutorijal: `https://www.w3schools.com/graphics/svg_intro.asp`



Rasterska grafika (jpg, png, gif, bmp, ...)

- Ekranu je pridružena matrica piksela – bitmapa
- Piksel – kvadrat određen celobrojnom mrežom (celobrojne tačke su centri kvadrata) ili krug sa centrom u čvorovima celobrojne mreže
- Koordinatni početak je u donjem levom uglu ekrana
- Svaki piksel ima jednu od 2^N vrednosti (N - broj bitova po pikselu)
- Memorija koju slika zauzima zavisi od rezolucije i bitske dubine
- Problemi pri skaliranju slike (nazubljenost, mutnost)
- Koristi se za fotografije, u veb dizajnu, ...

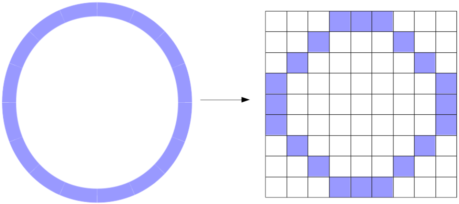


JPEG vs. PNG vs. GIF

- JPEG (Joint Photographic Expert Group)
  - Podržava 24-bitne boje
  - Dobar izbor za kompresovanje digitalnih slika visoke rezolucije
  - Kompresija je sa gubicima
  - Ne podržava transparentnu pozadinu (uvek su pravougaonog formata sa punom pozadinom)
  - Manja veličina datoteke (u odnosu na PNG) za sličan kvalitet slike
- PNG (Portable Network Graphic)
  - Podržava 24-bitne boje
  - Koristi kompresiju bez gubitaka
  - Veličina datoteke slika u visokoj rezoluciji je velika
  - Podržava transparentnu pozadinu (pogodna za logoe)
- GIF (Graphics Interchange Format)
  - Podržava samo 8-bitne boje
  - Osnovna prednost u odnosu na JPEG i PNG je da kompresuje digitalne slike u manje datoteke (smanjivanjem broja boja)
  - Podržava transparentnost i kratke animacije male veličine

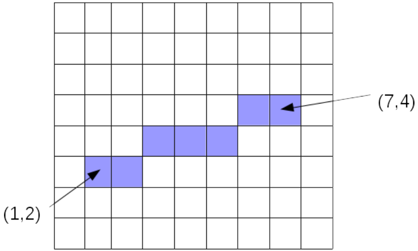
Rasterizacija

- Rasterizacija (ili sken konverzija) je proces transformisanja objekata niskog nivoa u njihovu rastersku verziju (tj. reprezentaciju pikselima)
- Podrazumeva se da nam je na raspolaganju operacija:
  - `setpixel(x,y)`



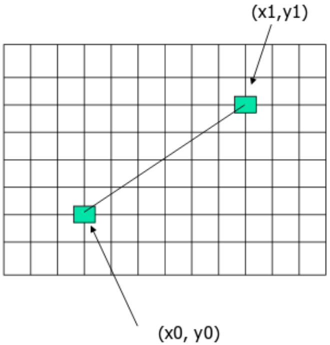
Crtanje duži – opis problema

- Zadatak: Postaviti boje piksela tako da aproksimiraju duž od tačke (x_0, y_0) do tačke (x_1, y_1)
- Zahtevi:
  - Što bliže idealnoj pravoj
  - Obavezno prolazi kroz krajnje tačke
  - Sve tačke svake duži su iste osvetljenosti bez obzira na nagib i dužinu
  - Crtanje treba da bude što brže

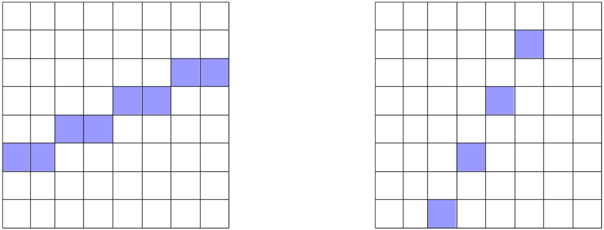


Jednačine prave

- Eksplicitna jednačina prave je $y = mx + B$
- Iz pripadnosti tačaka (x_0, y_0) i (x_1, y_1) pravoj, dobijamo $m = \frac{y_1 - y_0}{x_1 - x_0}$
- Jednačina prave kroz jednu datu tačku (x_0, y_0) je $y = m(x - x_0) + y_0$



Koeficijent pravca prave



- Ako bi se u svakoj koloni označavao po jedan piksel, bila bi vidljiva razlika u osvetljenosti duži različitog koeficijenta pravca
- Zato uvodimo naredno pravilo:
  - $|m| < 1$ – crta se po tačno jedan piksel u svakoj koloni
  - $m > 1$ ili $m < -1$ – crta se po tačno jedan piksel u svakoj vrsti

Crtanje duži – osnovni zadatak

- Jednobitna slika (piksel ima dve moguće vrednosti: 0 i 1)
- Duž je debljine 1
- Duž ima koeficijent pravca m: $|m| < 1$
  (za $|m| > 1$ radimo simetrično)
- Kriterijum ocene kvaliteta: minimalno rastojanje od idealne duži
- Zbog pogodosti računa piksel ćemo u algoritmima razmatrati u terminima krugova sa centrom u čvorovima celobrojne mreže
  - možemo koristiti termine "između dve tačke", "prava je bliža jednom od dva susedna piksela" ...

Algoritam grube sile

```
procedure Line_brute_force(x0, y0, x1, y1 : integer);
var
   x : integer;
   dx, dy, y, m : real;

begin
   dx := x1 - x0;
   dy := y1 - y0;
   m := dy/dx;
   for x := x0 to x1 do
     begin
       y := m * (x - x0) + y0;
       setpixel(x, round(y));
     end
end.
```

Algoritam grube sile – nedostaci

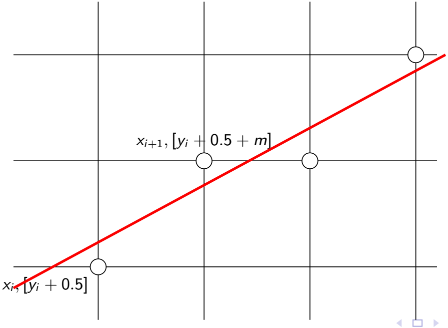
- Veliki broj operacija – svaki korak uključuje sabiranje, oduzimanje, množenje, zaokruživanje
- Korišćenje aritmetike u pokretnom zarezu – množenje, sabiranje, zaokruživanje
- Pomoćne promenljive su realnog tipa

Osnovni inkrementalni algoritam – ideja

- Važi veza:

$$y_{i+1} = mx_{i+1} + B = m(x_i + 1) + B = mx_i + m + B = y_i + m$$

- Ideja inkrementalnosti – za računanja u narednom koraku koristimo već izračunate vrednosti iz prethodnog koraka
- Na ovaj način eliminiše se množenje



Osnovni inkrementalni algoritam

```
procedure Line_increment_basic(x0, y0, x1, y1 : integer);
var
   x : integer;
   dx, dy, y, m : real;

begin
   dx := x1 - x0;
   dy := y1 - y0;
   m := dy/dx;
   y := y0;
   for x := x0 to x1 do
     begin
       setpixel(x, round(y));
       y := y + m
     end
end.
```

Osnovni inkrementalni algoritam – komentari

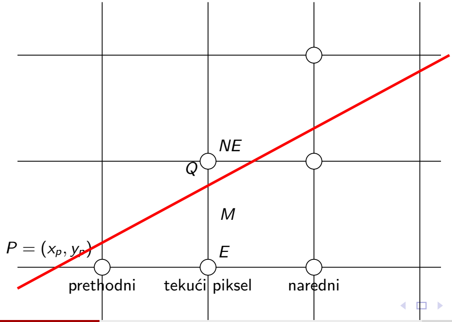
- Šta bi se desilo sa ovim algoritmom u slučaju $|m| > 1$?
- Nedostaci ovog algoritma:
  - m je realna vrednost izračunata sa nekom tačnošću i greška se nagomilava
  - I dalje vršimo zaokruživanje
  - Promenljive su i dalje realnog tipa

Midpoint algoritam (varijanta Bresenhamovog algoritma)

- Bresenham (1965) – crtanje duži korišćenjem celobrojne aritmetike
- Pitteway (1967), Van Aken (1984) – unapredili algoritam, može se uopštiti na proizvoljnu krivu drugog reda
- Zahtevi:
  - $0 < m < 1$
  - (x_0, y_0) je donja leva tačka, a (x_1, y_1) gornja desna i potrebno ih je spojiti

Midpoint algoritam (varijanta Bresenhamovog algoritma)

- Označili smo piksel $P(x_p, y_p)$ i u narednom koraku pravimo izbor između samo dva piksela: E i NE
  - Bresenham: razmatra rastojanje tačaka E i NE do presečne tačke duži koju rasterizujemo sa pravom $x = x_p + 1$ i bira bližu
  - Midpoint: razmatra odnos tačke M (središte duži određene sa E i NE) sa duži koju rasterizujemo: ako je M ispod preseka duži koju rasterizujemo sa pravom $x = x_p + 1$ biramo NE, a inače E



Midpoint algoritam (nastavak)

- Kako proveravamo da li se tačka M nalazi iznad ili ispod date prave korišćenjem celobrojne aritmetike?
- Iz eksplicitne jednačine prave dobijamo njenu implicitnu jednačinu:

$$y = \frac{dy}{dx}x + B \quad (dx = x_1 - x_0, dy = y_1 - y_0)$$

$$F(x, y) = dy \cdot x - dx \cdot y + B \cdot dx = 0, \quad -dx < 0$$

$$F(x, y) = a \cdot x + b \cdot y + c = 0, \quad a = dy, b = -dx < 0, c = B \cdot dx$$

- Vrednost $F(x, y)$ jednaka je 0 za tačke na pravoj, pozitivna je za tačke ispod prave, a negativna za tačke iznad prave

Midpoint algoritam – promenljiva odlučivanja

- Vrednost $d = F(x_p + 1, y_p + \frac{1}{2})$ na osnovu koje se pravi izbor zovemo promenljiva odlučivanja:
  - $d < 0$ – biramo E
  - $d > 0$ – biramo NE
  - $d = 0$ – stvar dogovora, biramo E
- Koja je pozicija tačke M i vrednost promenljive d za narednu liniju mreže? Odgovor zavisi od toga da li smo u prethodnom koraku odabrali piksel E ili NE

Midpoint algoritam – ažuriranje promenljive odlučivanja

- Ako smo u prethodnom koraku odabrali E:

$$d_{new} = F(x_p + 2, y_p + \frac{1}{2}) = a(x_p + 2) + b(y_p + \frac{1}{2}) + c =$$

$$a + a(x_p + 1) + b(y_p + \frac{1}{2}) + c = a + d_{old} = d_{old} + dy = d_{old} + \Delta_E$$

- Ako smo u prethodnom koraku odabrali NE:

$$d_{new} = F(x_p + 2, y_p + \frac{3}{2}) = a(x_p + 2) + b(y_p + \frac{3}{2}) + c =$$

$$a+b+a(x_p+1)+b(y_p+\frac{1}{2})+c = a+b+d_{old} = d_{old}+dy-dx = d_{old}+\Delta_{NE}$$

- Δ_E, odnosno Δ_{NE} je korektivni faktor i nazivamo ga razlikom unapred

Midpoint algoritam – početna iteracija

- Algoritam u svakom koraku bira između dva piksela na osnovu znaka promenljive odlučivanja
- Naredna vrednost promenljive odlučivanja računa se na osnovu prethodne – inkrementalni pristup
- Potrebno nam je da eksplicitno izračunamo prvu vrednost
- Za prvu tačku važi:

$$d_{start} = F(x_0 + 1, y_0 + \frac{1}{2}) = a(x_0 + 1) + b(y_0 + \frac{1}{2}) + c = a + b/2$$

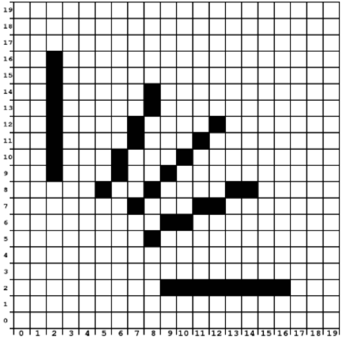
- Da bi se izbeglo deljenje sa dva, sve vrednosti množimo sa 2; pritom relevantni znakovi ostaju isti
  - $d_{start} = 2a + b = 2dy - dx$
  - ako je $d_{old} \leq 0$, onda $d_{new} = d_{old} + 2dy$
  - ako je $d_{old} > 0$, onda je $d_{new} = d_{old} + 2dy - 2dx$

Midpoint algoritam za crtanje duži

```
procedure Line_midpoint(x0, y0, x1, y1 : integer);
var
   dx, dy, x, y, f : integer;
begin
   dx := x1 - x0;
   dy := y1 - y0;
   y := y0;
   d := 2*dy - dx;
   for x := x0 to x1 do
     begin
       setpixel(x, y);
       if (d <= 0)
         begin
           d := d + 2*dy;
         end
       else
         begin
           y := y+1;
           d := d + 2*(dy-dx);
         end
     end
end.
```

Crtanje duži – dodatna pitanja

- Duži različitog nagiba a iste dužine imaju različite osvetljenosti



- Duži $P_0 P_1$ i $P_1 P_0$ treba da se crtaju na isti način
  - zbog stilova nije dobro uvek svoditi na poredak sleva nadesno
  - važan je odabir piksela kada je vrednost promenljive odlučivanja $d = 0$

Crtanje kruga – opis problema

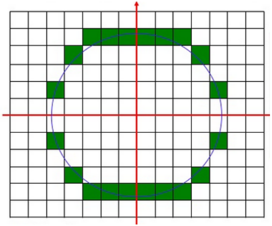
- Zadatak: Postaviti boje piksela tako da aproksimiraju sliku kruga sa centrom u $(0, 0)$
- Zahtevi:
  - Što bliže idealnom krugu
  - Crtanje treba da bude što brže

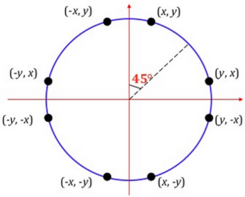
Crtanje kruga – osnovni zadatak

- Jednobitna slika (piksel ima dve moguće vrednosti: 0 i 1)
- Krug je debljine 1
- Centar kruga je u tački $(0, 0)$
- Radimo sa jednim kvadrantom; na osnovu simetrije mogu se odrediti pikseli u ostalim kvadrantima

Algoritam grube sile

- Implicitna jednačina kruga je: $x^2 + y^2 = R^2$, a iz nje možemo dobiti eksplicitnu jednačinu: $y = \pm\sqrt{R^2 - x^2}$
- x_i se počev od vrednosti 0 inkrementira za po 1 i osvetljavamo tačku: $(x_i, [\sqrt{R^2 - x_i^2} + 0.5]),\ i = 1, 2, 3, \ldots$
- Nedostaci:
  - neefikasno: aritmetika u pokretnom zarezu, računanje korena, zaokruživanje, ...
  - sa porastom vrednosti x povećavaju se praznine između tačaka

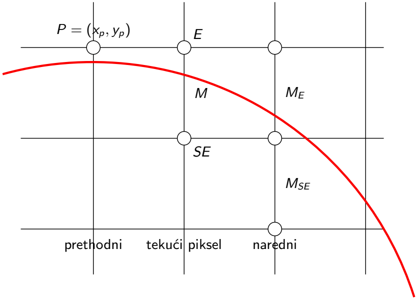




Midpoint (Bresenham-ov) inkrementalni algoritam

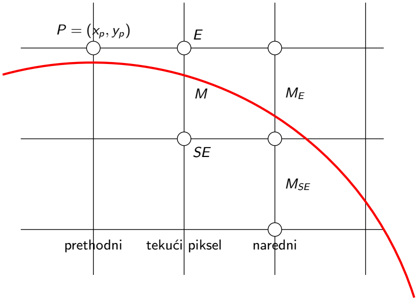
- Bresenham (1977), Midpoint algoritam je varijanta Bresnehamovog algoritma
- Razmatramo samo osminu kruga; ostali slučajevi obrađuju se simetrično
- Vrednost x ide od 0 do $R/\sqrt{2}$
- Osnovna ideja slična je ideji za crtanje duži: u svakom koraku treba odabrati jednu od dve moguće tačke

Midpoint (Bresenham-ov) inkrementalni algoritam



- Vrednost $F(x,y) = x^2 + y^2 - R^2$ je jednaka 0 u tačkama koje pripadaju krugu, pozitivna je u spoljašnjosti, a negativna u unutrašnjosti kruga
- Ako je tačka M u unutrašnjosti kruga, onda je tačka E bliža krugu nego tačka SE. Ako je tačka M u spoljašnjosti kruga, onda je tačka SE bliža krugu

Midpoint algoritam – promenljiva odlučivanja



- Vrednost $d = F(x_p + 1, y_p - \frac{1}{2}) = (x_p + 1)^2 + (y_p - \frac{1}{2})^2 - R^2$ na osnovu koje se pravi izbor nazivamo promenljiva odlučivanja:
  - $d < 0$, bira se tačka E
  - $d > 0$, bira se tačka SE
  - $d = 0$, stvar dogovora, bira se tačka SE

Midpoint algoritam – ažuriranje promenljive odlučivanja

- Ako je $d_{old} < 0$, tačka E je izabrana i važi:

$$d_{new} = F(x_p+2, y_p-\frac{1}{2}) = (x_p+2)^2+(y_p-\frac{1}{2})^2-R^2 = d_{old}+(2x_p+3)$$

- Ako je $d_{old} \geq 0$, tačka SE je izabrana i važi:

$$d_{new} = F(x_p+2, y_p-\frac{3}{2}) = (x_p+2)^2+(y_p-\frac{3}{2})^2-R^2 = d_{old}+(2x_p-2y_p+5)$$

- U slučaju $d_{old} < 0$ koristi se $\Delta_E = 2x_p + 3$
- U slučaju $d_{old} \geq 0$ koristi se $\Delta_{SE} = 2x_p - 2y_p + 5$
- Obe ove razlike zavise od koordinata tačke P

Midpoint algoritam za crtanje kruga – analiza

- Početni uslov (za celobrojni poluprečnik):

$$F(0, R) = 0$$

$$F(1, R - \frac{1}{2}) = 1 + (R^2 - R + \frac{1}{4}) - R^2 = \frac{5}{4} - R$$

- Prva vrednost za d nije celobrojna
- Koristićemo zamenu $h = d - \frac{1}{4}$: tada je inicijalna vrednost $h = 1 - R$, a poređenje $d < 0$ postaje $h < -\frac{1}{4}$
- Kako je inicijalna vrednost promenljive d celobrojna i kako su vrednosti koje se dodaju (Δ_E i Δ_{SE}) celobrojne, ovaj uslov može da se zameni sa $h < 0$
- Ovako uvedenu promenljivu h na kraju označavamo sa d

Midpoint algoritam za crtanje kruga

```
procedure MidpointCircle (r : integer);
var
   x, y, d : integer;
begin
   x := 0;
   y := r;
   d := 1 - r;
   setpixel(x, y);

   while (y > x) do
      begin
         if d < 0 then    { select E }
            begin
               d := d + 2*x + 3;
               x := x + 1;
            end
         else       { select SE }
            begin
               d := d + 2*(x-y) + 5;
               x := x + 1;
               y := y - 1;
            end
         setpixel(x, y);
      end
end.
```

Unapređeni midpoint algoritam za crtanje kruga

- Vrednosti Δ_E i Δ_{SE} su linearne te za njihovo izračunavanje takođe možemo koristiti tehniku inkrementalnog uvećavanja
- Ideja je da se i za ove vrednosti, kao kod razlika unapred, nove vrednosti Δ_{Enew} i Δ_{SEnew} izraze preko starih Δ_{Eold} i Δ_{SEold}
- Pritom u svakom koraku moramo ažurirati i Δ_E i Δ_{SE} jer nam ažurirana verzija neke od ovih vrednosti može zatrebati u nekoj narednoj iteraciji
- Prilikom analize, važnu ulogu igra informacija koja je tačka bila izabrana u prethodnom koraku, da bismo znali gde se premešta naredna referentna tačka
- Vrednosti za koje se uvećavaju promenljive Δ_E i Δ_{SE} nazivamo razlike drugog reda

Unapređeni midpoint algoritam – ažuriranje razlika drugog reda

- Ako je izabrana tačka E, referentna tačka se pomera iz (x_p, y_p) u $(x_p + 1, y_p)$
  - Vrednost Δ_{Eold} u (x_p, y_p) je jednaka $2x_p + 3$, a vrednost Δ_{Enew} (u tački $(x_p + 1, y_p)$) je jednaka

$$\Delta_{Enew} = 2(x_p + 1) + 3 = \Delta_{Eold} + 2$$

  - Vrednost Δ_{SEold} u (x_p, y_p) jednaka je $2x_p - 2y_p + 5$, a vrednost Δ_{SEnew} (u tački $(x_p + 1, y_p)$) je jednaka

$$\Delta_{SEnew} = 2(x_p + 1) - 2y_p + 5 = \Delta_{SEold} + 2$$

Unapređeni midpoint algoritam – ažuriranje razlika drugog reda

- Ako je izabrana tačka SE, referentna tačka se pomera iz (x_p, y_p) u $(x_p + 1, y_p - 1)$
  - Vrednost Δ_{Eold} u (x_p, y_p) je jednaka $2x_p + 3$, a vrednost Δ_{Enew} (u tački $(x_p + 1, y_p - 1)$) je jednaka

$$\Delta_{Enew} = 2(x_p + 1) + 3 = \Delta_{Eold} + 2$$

  - Vrednost Δ_{SEold} u (x_p, y_p) jednaka je $2x_p - 2y_p + 5$, a vrednost Δ_{SEnew} (u tački $(x_p + 1, y_p - 1)$) je jednaka

$$\Delta_{SEnew} = 2(x_p + 1) - 2(y_p - 1) + 5 = \Delta_{SEold} + 4$$

Unapređeni midpoint algoritam – ažuriranje razlika drugog reda

- Početne vrednosti za Δ_E i Δ_{SE} se dobijaju za polaznu tačku $(x_p, y_p) = (0, R)$
  - $\Delta_{E_{start}} = 2x_p + 3 = 3$
  - $\Delta_{SE_{start}} = 2x_p - 2y_p + 5 = -2R + 5$

Unapređeni midpoint algoritam za crtanje kruga

```
procedure MidpointCircle2 (r : integer);
var
   x, y, d, deltaE, deltaSE : integer;
begin
   x := 0;
   y := r;
   d := 1 - r;
   deltaE := 3;
   deltaSE := -2*r + 5;
   setpixel(x, y);

   while (y > x) do
      begin
         if d < 0 then    { select E }
            begin
               d := d + deltaE;
               deltaE := deltaE + 2;
               deltaSE := deltaSE + 2;
               x := x + 1;
            end
         else       { select SE }
            begin
               d := d + deltaSE;
               deltaE := deltaE + 2;
               deltaSE := deltaSE + 4;
               x := x + 1;
               y := y - 1;
            end
         setpixel(x, y);
      end
end.
```

Crtanje kruga čiji centar nije u koordinatnom početku

- Ako središte kruga nije tačka $(0,0)$ nego tačka (a,b), koristi se algoritam sličan navedenom
- Nije isplativo za svaki piksel pojedinačno dodavati a i b (u odnosu na piksele kruga sa središtem $(0,0)$)
- Sve razlike prvog i drugog reda su iste (ako je poluprečnik isti) bez obzira na središte kruga
- Razlikuje se samo početni piksel – umesto $(0,R)$ prvi uključeni piksel treba da bude $(a,R+b)$, a umesto uslova $y>x$ treba koristiti uslov $y-b>x-a$, tj. efikasnije $y>x+(b-a)$