Kružnica

Algoritmus na vykreslenie kružnice z rovnice kružnice v karteziánskych súradniciach

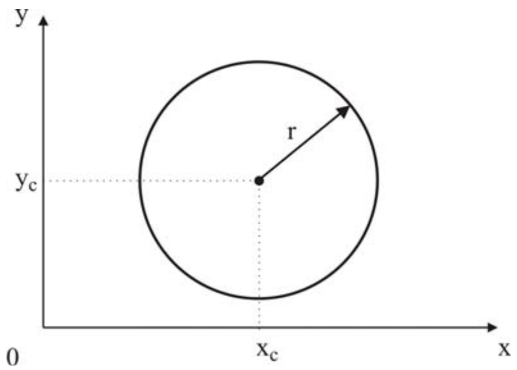
- kružnica je najčastejšie definovaná pomocou stredu a polomeru (obr. 1)
- jej rovnicu môžeme napísať v karteziánskych súradniciach

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

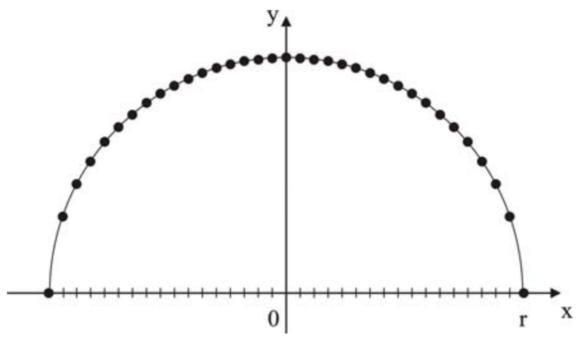
- táto rovnica sa dá použiť na vykreslenie kružnice postupnou zmenou súradnice x o jednotku od $x_c - r$ do $x_c + r$ a výpočtom zodpovedajúcich hodnôt y pre každé x

$$y = y_c \pm \sqrt{r^2 - (x - x_c)^2}$$

- tento postup vyžaduje zložité výpočty v každom kroku
- rozmiestnenie bodov na kružnici nie je pravidelné (obr. 2) – môžeme ho odstrániť zámenou x a y – postupne budeme meniť súradnicu y o jednotku a vypočítavať zodpovedajúce x



Obr. 1 Charakteristické hodnoty potrebné na zadanie kružnice



Obr. 2 Nerovnomerne generovaná kružnica pri konštantnom kroku x

Algoritmus na vykreslenie kružnice z rovnice kružnice v polárnych súradniciach

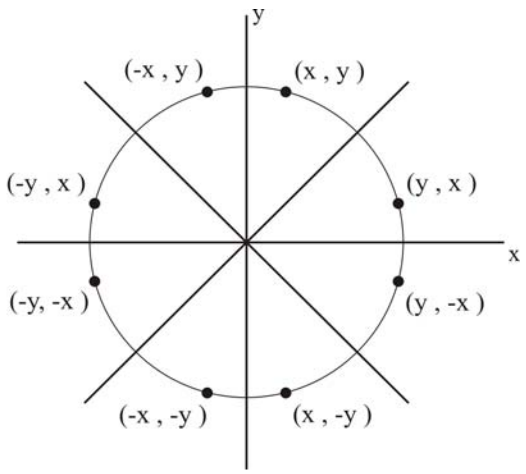
- ďalším spôsobom odstránenia nerovnomerného rozloženia bodov je výpočet ich pozícií pomocou polárnych súradníc

$$x = x_c + r \cos \alpha$$

$$y = y_c + r \sin \alpha \qquad \alpha \in \langle 0, 2\pi \rangle$$

- pri zmene α o konštantnú veľkosť budú body na kružnici vykreslené v pravidelných vzdialenostiach
- prírastok α závisí od aplikácie

- najčastejšie sa volí $1/r$, vtedy sú generované body so súradnicami približne vo vzdialenosti 1 pixel túto metódu môžeme vylepšiť použitím symetrie kružnice (obr. 3)
- z daného bodu na kružnici môžeme odvodiť ďalšie body zámenou súradníc a zmenou ich znamienka
- teda celú kružnicu môžeme nakresliť tak, že vypočítame súradnice bodov od $x = 0$ po $x = y$ a ostatné body odvodíme použitím symetrie kružnice (obr. 3)
- doteraz uvedené metódy na výpočet súradníc pixlov na kružnici sú náročné na výpočet
- oveľa výhodnejšie je použiť metódy na výpočet súradníc bodov v celočíselnej aritmetike



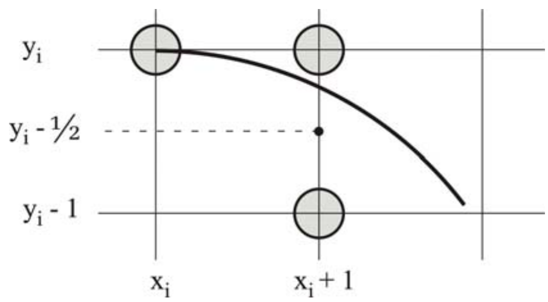
Obr. 3 Využitie symetrie pri vykreslení kružnice

Algoritmus na vykreslenie kružnice v celočíselnej aritmetike (Bresenhamov algoritmus)

- podobne ako v algoritme na vykreslenie úsečky, aj pri kružnici sa dajú určiť celočíselné súradnice pixlov tak, že v každom kroku porovnáme, ktorý z dvoch pixlov leží bližšie ku skutočnej kružnici
- uvažujme najprv o kružnici so stredom v začiatku sústavy súradníc ($x_c = 0$, $y_c = 0$)
- budeme počítať súradnice bodov v $1/8$ kružnice a pomocou súmernosti (obr. 3) vykreslíme jej zvyšok
- začneme s $x = 0$, konštantné prírastky budeme pridávať v smere osi x a skončíme hodnotou $x = y$
- algoritmus je založený na tom istom princípe ako Bresenhamov algoritmus na vykreslenie úsečky
- ak má kružnica stred v bode $(0 , 0)$ potom jej rovnica je $x^2 + y^2 = r^2$
- označíme funkciu $F (x , y) = x^2 + y^2 - r^2$

Algoritmus:

- predpokladáme, že bod (x_i , y_i) je už vykreslený



Obr. 4 Časť kružnice v rastri

- nasledujúci bod môže byť $(x_i + 1 , y_i)$ alebo $(x_i + 1 , y_i - 1)$
- na obr. 4 je naznačený bod $(x_i +1, y_i -1/2)$, ktorý sa nachádza v strede medzi uvedenými bodmi
- ak dosadíme jeho súradnice do funkcie F, potom znamienko určí, či tento bod leží vo vnútri alebo zvonka kružnice

  $p_i = F (x_i + 1, y_i - 1/2) = (x_i + 1)^2 + (y_i - 1/2)^2 - r^2$
- ak $p_i < 0$, potom bod leží vo vnútri kružnice, teda bližšie ku skutočnej kružnici je bod $(x_i + 1, y_i)$
- ak $p_i \geq 0$, potom bod leží zvonka kružnice, teda bude vybraný bod $(x_i + 1, y_i - 1)$
- vypočítame p_{i+1} z p_i tak, že namiesto i dosadíme $i+1$
- vypočítame rozdiel $p_{i+1} - p_i$ a z neho vyjadríme p_{i+1} pomocou p_i

začneme s výpočtom p_1

- *ak* $p_i \geq 0$:

$$x_{i+1} = x_i + 1$$

$$y_{i+1} = y_i - 1$$

$$p_{i+1} = (x_{i+1} + 1)^2 + (y_{i+1} - 1/2)^2 - r^2 = (x_i + 2)^2 + (y_i - 3/2)^2 - r^2$$

$$p_{i+1} - p_i = (x_i + 2)^2 + (y_i - 3/2)^2 - r^2 - ((x_i + 1)^2 + (y_i - 1/2)^2 - r^2) =$$

$$= x_i^2 + 4x_i + 4 - (x_i^2 + 2x_i + 1) + y_i^2 - 3y_i + \frac{9}{4} - (y_i^2 - y_i + \frac{1}{4}) -$$

$$- r^2 + r^2 = 2x_i + 3 - 2y_i + 2$$

$$p_{i+1} = p_i + 2x_i - 2y_i + 5$$

- *ak* $p_i < 0$:

$$x_{i+1} = x_i + 1$$

$$y_{i+1} = y_i$$

$$p_{i+1} = (x_{i+1} + 1)^2 + (y_{i+1} - 1/2)^2 - r^2 = (x_i + 2)^2 + (y_i - 1/2)^2 - r^2$$

$$p_{i+1} - p_i = (x_i + 2)^2 + (y_i - 1/2)^2 - r^2 - ((x_i + 1)^2 + (y_i - 1/2)^2 - r^2) =$$

$$= x_i^2 + 4x_i + 4 - (x_i^2 + 2x_i + 1) + y_i^2 - y_i + \frac{1}{4} - (y_i^2 - y_i + \frac{1}{4}) -$$

$$- r^2 + r^2 = 2x_i + 3$$

$$p_{i+1} = p_i + 2x_i + 3$$

- tieto vzťahy sú podobné ako vzťahy pre úsečku vykreslovanú Bresenhamovým algoritmom
- obsahujú len sčítanie a násobenie dvoma
- násobenie je spomaľujúcim faktorom - dá sa odstrániť
- aby sme nemuseli pri každom novom výpočte premennej p násobiť aktuálne súradnice x_i a y_i dvojkou, sú v algoritme zavedené premenné *dx* a *dy*
- *dx* obsahuje dvojnásobok x_i
- *dy* obsahuje dvojnásobok y_i
- nové *dx* a *dy* počítame pričítavaním alebo odčítavaním *2*, pretože hodnoty x_i a y_i sa menia o *1*

$p_1 = (x_1 + 1)^2 + (y_1 - 1/2)^2 - r^2 = (0 + 1)^2 + (r - 1/2)^2 - r^2 = 1 + r^2 - r + 1/4 - r^2 = 5/4 - r$

- procedúra Kresli_symetricke_body nakreslí *8* pixlov usporiadaných symetricky
- pri jej použití môže prísť k násobnému kresleniu pixlov s rovnakými súradnicami, napr. vtedy, keď sú symetrické body odvodzované z bodu (*0 , r*) - teda procedúra musí obsahovať testy na rovnosť súradníc kreslených pixlov

```
procedure Kruznica (x_stred , y_stred , polomer : integer);
var     p , dx , dy , x , y : integer;

        procedure Kresli_symetricke_body;
                begin
                        Putpixel (x_stred + x , y_stred + y);
                        Putpixel (x_stred - x , y_stred + y);
                        Putpixel (x_stred + x , y_stred - y);
                        Putpixel (x_stred - x , y_stred - y);
                        Putpixel (x_stred + y , y_stred + x);
                        Putpixel (x_stred - y , y_stred + x);
                        Putpixel (x_stred + y , y_stred - x);
                        Putpixel (x_stred - y , y_stred - x);
                end ;

begin
        x := 0;
        y := polomer;
        p := 1 - polomer;
        dx := 3;
        dy := 2*polomer - 2;
        repeat
                Kresli_symetricke_body;
                if p ≥ 0 then
                                begin
                                        p := p - dy;
                                        dy := dy - 2;
                                        y := y - 1;
```

```
                    end ;
            p := p + dx;
            dx := dx + 2;
            x := x + 1;
        until x > y;
end;
```