Generování základních objektů v rastru
2. cvičení předmětu IZG

Martin Velas

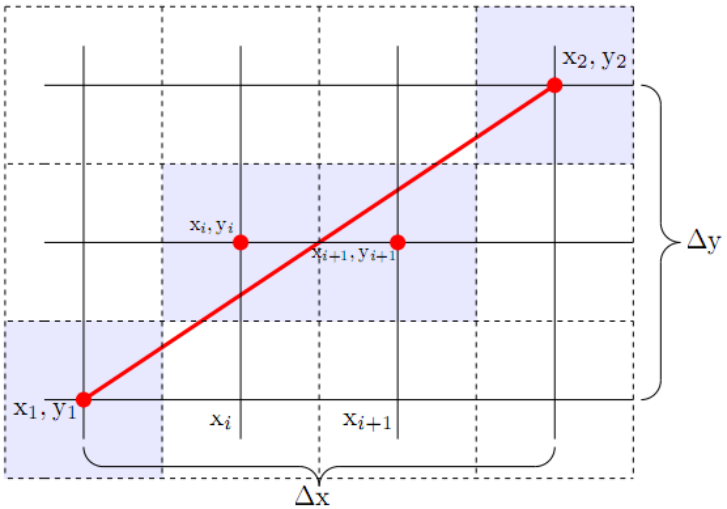
ivelas@fit.vutbr.cz

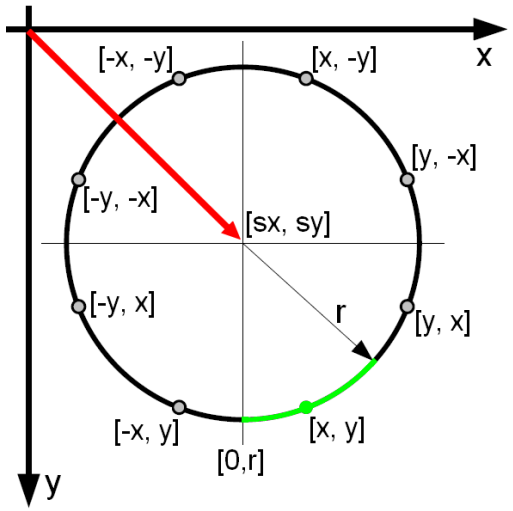
(s využitím materiálov M. Španěla a M. Šolonyho)

6. februára 2015

Náplň cvičenia

1. rasterizácia *úsečky*
   - prehľad algoritmov
   - samostatná úloha – DDA algoritmus s fixed point aritmetikou
     - 2 body
2. rasterizácia *kružnice*
   - samostatná úloha – MidPoint algoritmus
     - 1 bod

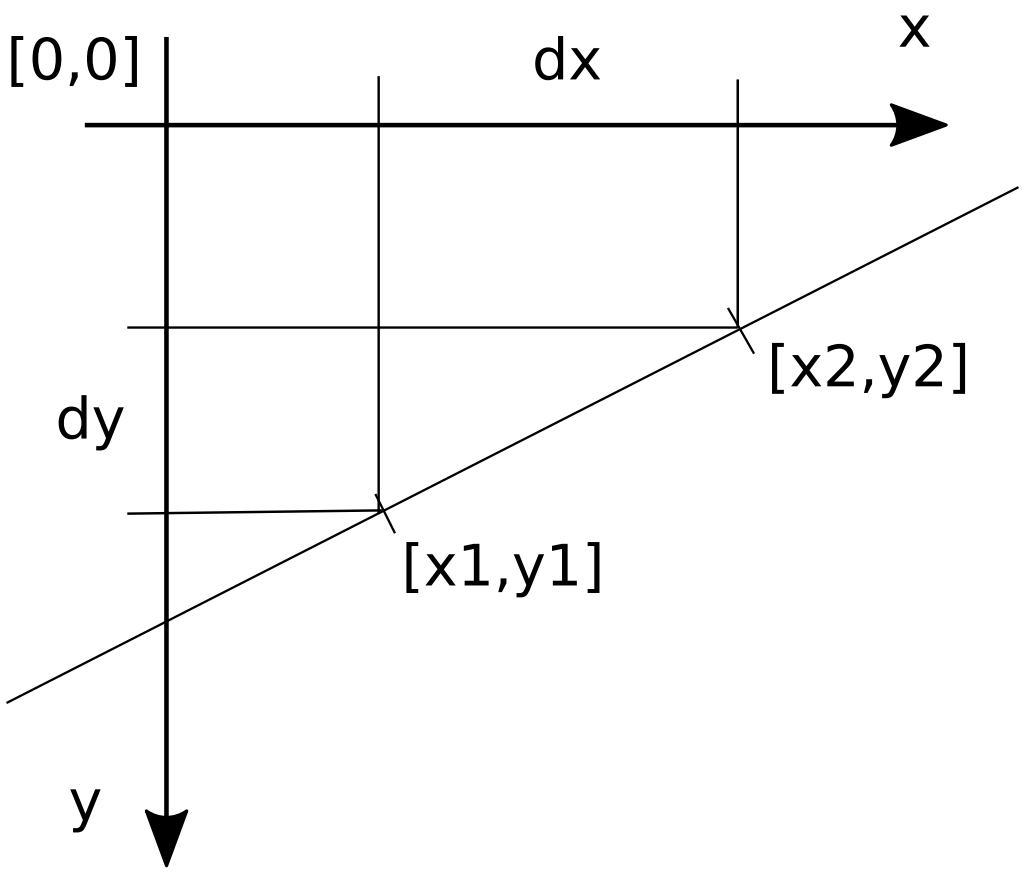




Popis priamky v rovine

?

Popis priamky v rovine



Smernicový popis priamky

- $y = kx + q$
- smernica $k = \frac{dy}{dx} = \tan(\alpha)$

Algoritmy

Algoritmy rasterizácie úsečky

- DDA ✗
- DDA s kontrolovaním chyby ✗
- Bresenhamov algoritmus ✓
- DDA s fixed point aritmetikou ✓

DDA - Digital Differential Analyser

Vlastnosti

- jednoduchý
- používa float point aritmetiku ⇒ neefektívny

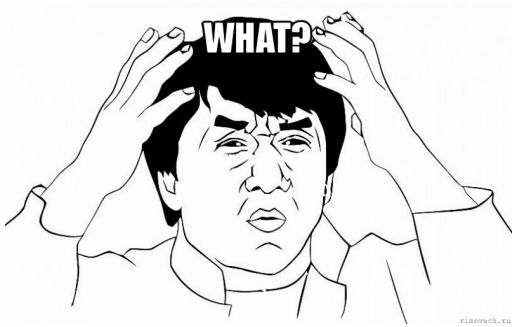
```
LineDDA(int x1, int y1, int x2, int y2) {
    const double k = (y2-y1) / (x2-x1);
    double y = y1;
    for(int x = x1; x <= x2; x++) {
        putPixel(x, y);     // ROUND
        y += k;
    }
}
```

- modifikácia: DDA s kontrolovaním chyby – stále float

Bressenhamov algoritmus

```
LineBressenham(int x1, int y1, int x2, int y2) {
    const int dx = x2-x1 , dy = y2-y1;
    const int P1 = 2*dy, P2 = P1 - 2*dx;
    int P = 2*dy - dx;
    int y = y1;
    for (int x = x1 ; x <= x2 ; x++) {
        putPixel(x, y);
        if (P >= 0) { P += P2 ; y++; }
        else        { P += P1 ; }
}
```

- *Prediktor* $\Rightarrow$ len celé čísla, efektívny, ale ...



DDA s fixed point aritmetikou

Zlatý Grál rasterizácie úsečky

- kódovanie do celých čísel (fixed point)
- rýchly, efektívny, používaný v GPU
- jednoduchý (takmer)

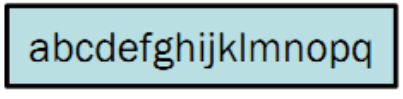
```
#define FRAC_BITS 8
LineDDAFixedPoint(int x1 , int y1 , int x2 , int y2
    int y = y1 << FRAC_BITS;
    const int k = ((y2-y1) << FRAC_BITS) / (x2-x1);

    for(int x = x1; x <= x2; x++) {
        putPixel(x, y >> FRAC_BITS);
        y += k;
    }
}
```

DDA s fixed-point aritmetikou

Floating-point aritmetika

- S . . . znaménko
- M . . . mantisa
- E . . . exponent

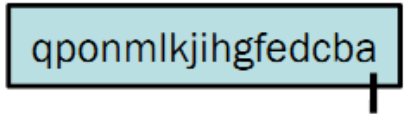
$$X = S \cdot M \cdot 2^E$$



=0,abcdefghijklmnopq

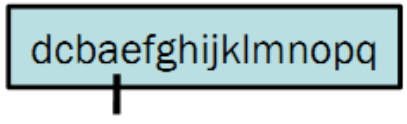
=a*1/2 + b*1/4 + c*1/8 + d*1/16 + …

Fixed-point aritmetika



=qponmlkjihgfedcba,0

=a*1 + b*2 + c*4 + d*8 + …



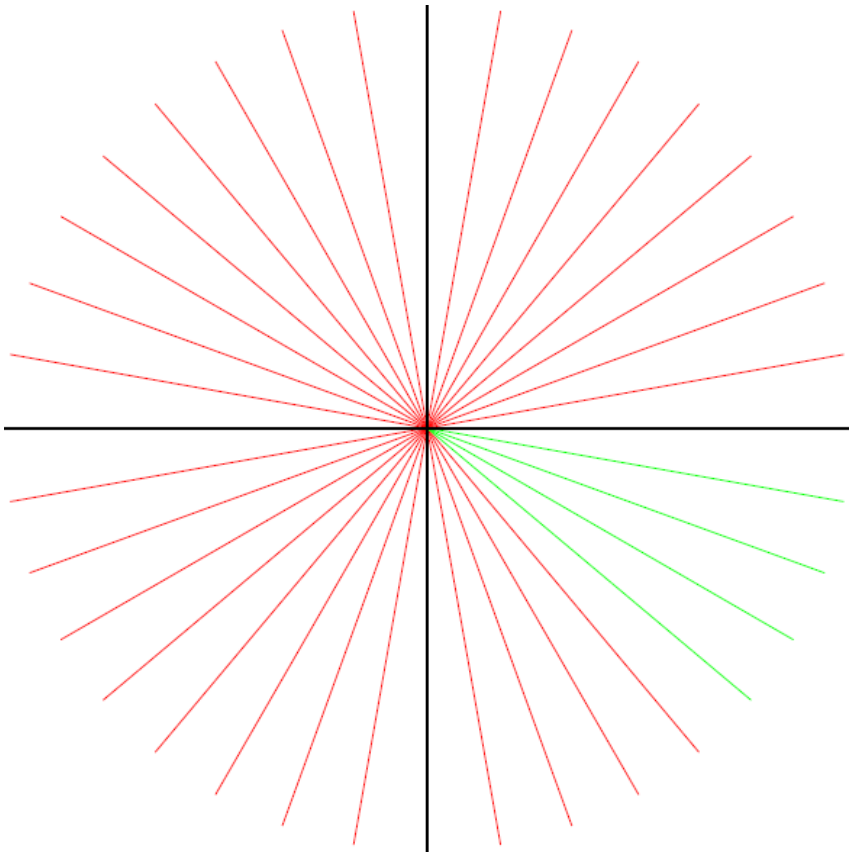
=dcba,efghijklmnopq

=a*1 + b*2 + c*3 + d*4 +
e*1/2 + f*1/4 + g*1/8 + h*1/16 + …

Problém uvedených algoritmov

Problém

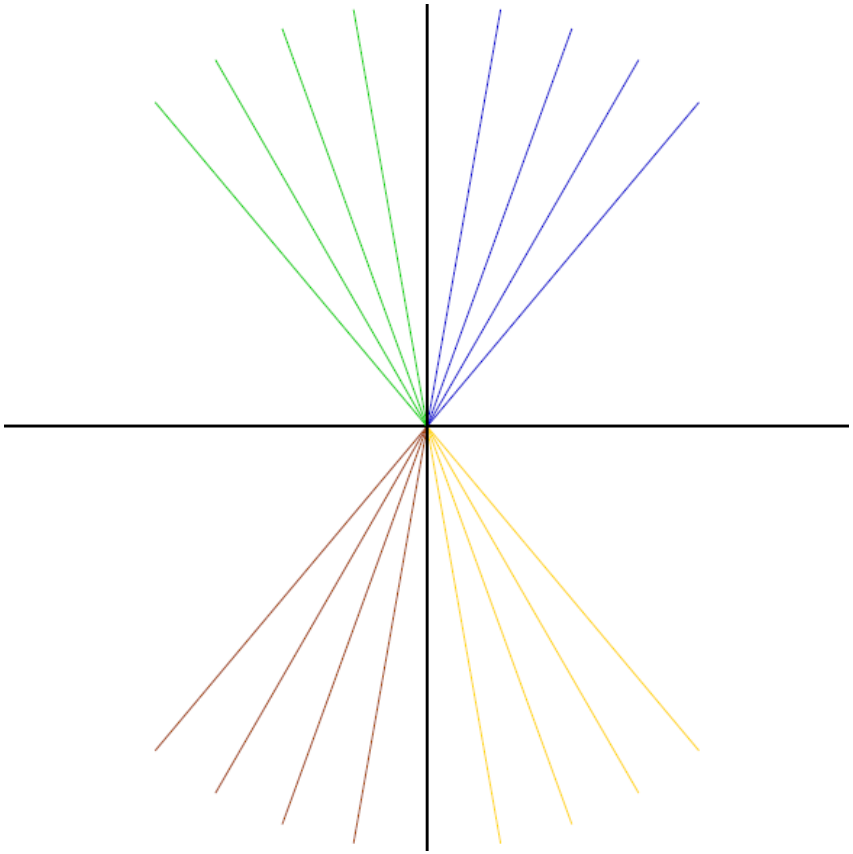
- vykreslenie úsečky v ľubovoľnom smere
- väčšina len v prvej polovici *I*. kvadrantu
- zadanie prvej bodovanej úlohy



Úsečka rýchlo rastie/klesá

$||\mathbf{dy}|| > ||\mathbf{dx}||$

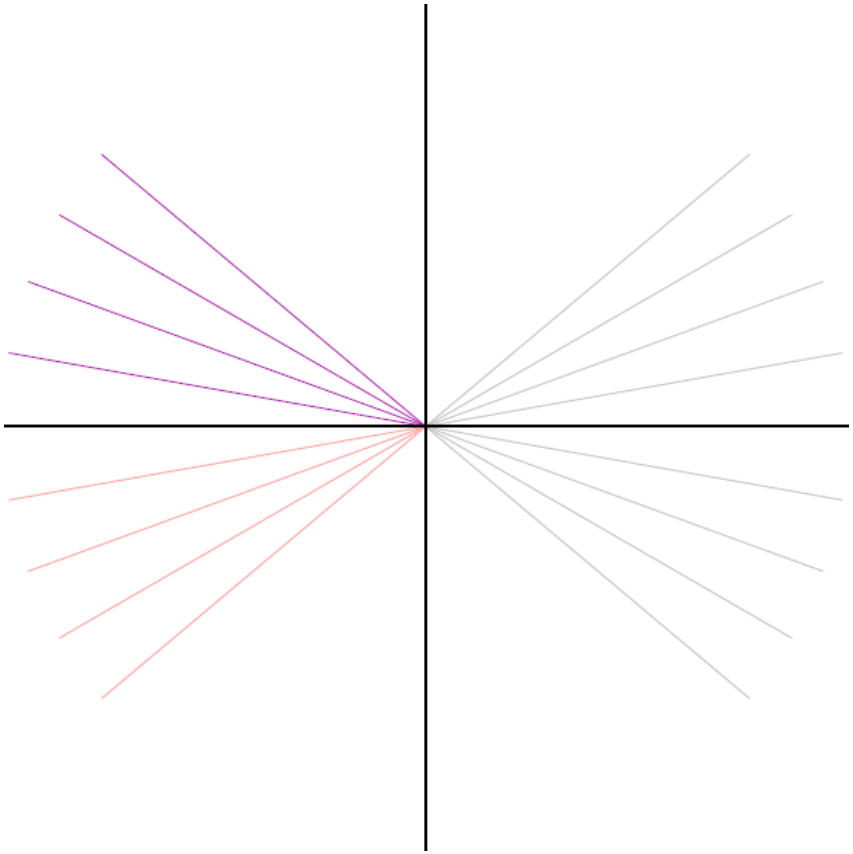
- výmena X-ových a Y-ových súradníc
- $x1 \Leftrightarrow y1$ a $x2 \Leftrightarrow y2$
- znovu vymeniť pri zápise do framebufferu (`putPixel`)



Úsečka smeruje sprava doľava

$x_1 > x_2$

- výmena počiatočného a koncového bodu
- $x1 \Leftrightarrow x2$ a $y1 \Leftrightarrow y2$
- zvyšok bez zmeny



Počiatočný bod zhodný s koncovým

[x1, y1] = [x2, y2]

- toto nie je úsečka, ale len 1 bod
- `Division by zero!`

Úloha 1 (2b)

Oprava funkcie `drawLine`

- v súbore `student.cpp`
- vykreslí kružnicu algoritmom DDA s FX aritmetikou každým smerom
- nezhavaruje ☺

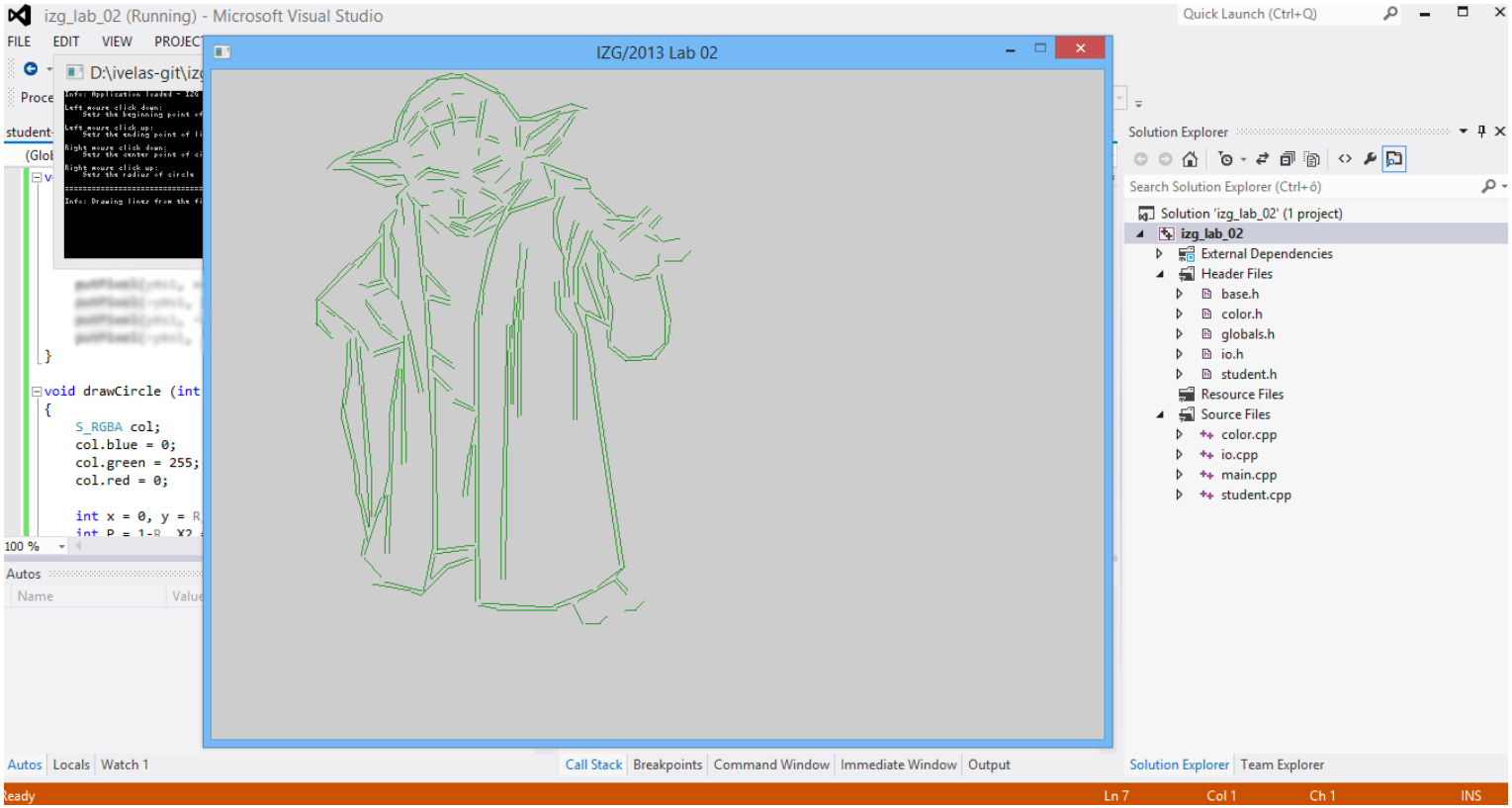
Tipy

- vykreslenie ľavým tlačítkom myši
- makro `SWAP(a, b)` pre výmenu, `putPixel(x, y, color)` pre zápis do FB
- dodržujte poradie uvedených úprav

- a ak bude všetko fungovať ...

The right solution find, you have!

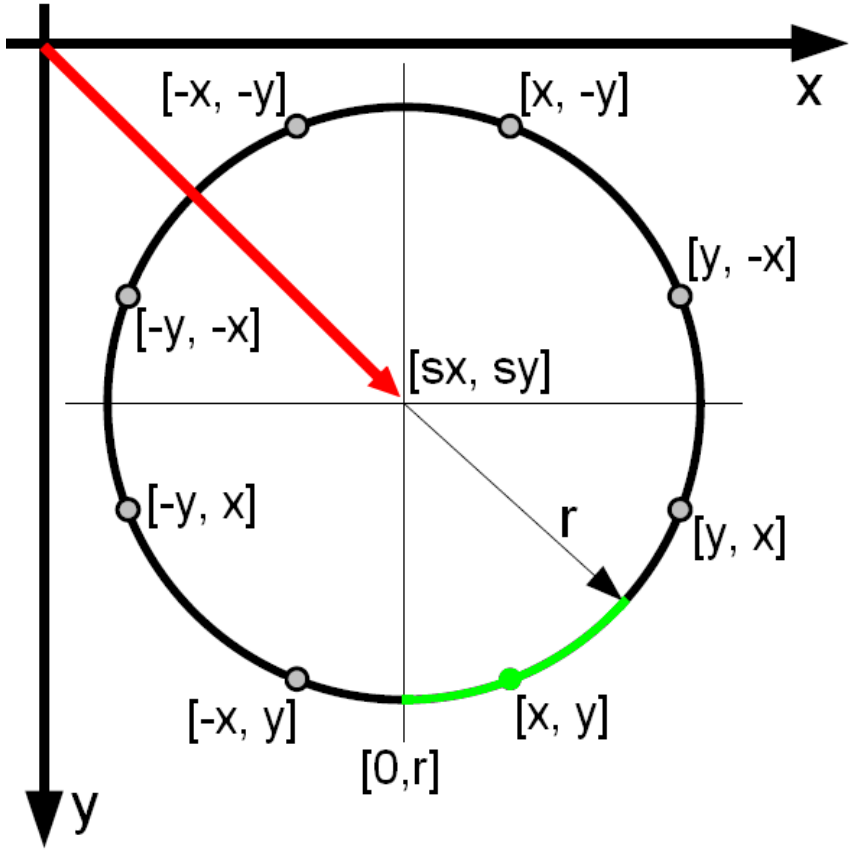
- ... a máte predstavivosť, vykreslí sa Master Yoda
- klávesa "D"



Popis kružnice v rovine

?

Popis kružnice v rovine



Všeobecná rovnica kružnica v rovine

- $(x - s_x)^2 + (y - s_y)^2 = r^2$
- stred kružnice $[s_x, s_y]$ a polomer r

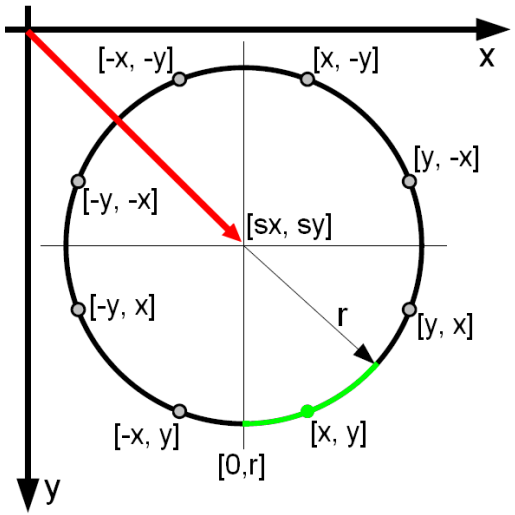
MidPoint algoritmus

Vlastnosti

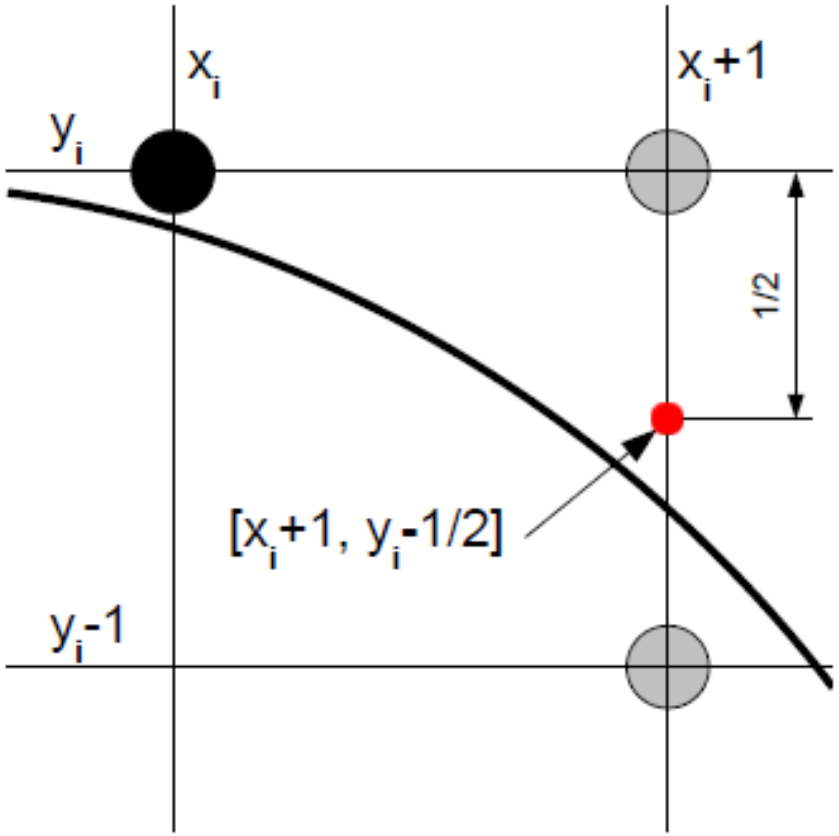
- alá Bresenhamov algoritmus
- prediktor, celočíselná aritmetika

Algoritmus

- od bodu $[0, r]$, kým $x <= y$
- $x++$, posun $y--$ podľa prediktoru



Testovanie MidPointu



midpoint vnútri kružnice $\Rightarrow \quad P < 0 \Rightarrow \quad y_{i+1} = y_i$

midpoint mimo kružnice $\Rightarrow \quad P \geq 0 \Rightarrow \quad y_{i+1} = y_i - 1$

MidPoint algoritmus - kód

```
void drawCircleMidpoint(int sx, int sy, int R) {
    int x = 0, y = R;
    int P = 1-R, X2 = 3, Y2 = 2*R-2;
    while (x <= y) {
        put8PixelsOfCircle(x, y, sx, sy);
        if (P >= 0) {
            P += -Y2;
            Y2 -= 2;
            y--;
        }
        P += X2;
        X2 += 2;
        x++;
    }
}
```

Úloha 2 (1b)

Doplňte telo funkcie `put8PixelsOfCircle()`

- v súbore `student.cpp`
- vykreslenie bodov na všetkých osminách kružnice
- osová súmernosť $\Rightarrow$ zámena znamienok
- stred kružnice $[s1, s2] \Rightarrow$ posun každého vykresľovaného bodu

Tipy

- vykreslenie kružnice pomocou ľavého tlačítka myši
- použite funkciu `putPixel()` a makro `SWAP`

Výskumná skupina Robo@FIT

- `http://www.fit.vutbr.cz/research/groups/robo/`
- robotika, počítačové videnie, HMI, AI, senzory, ...
- SW $\ggg$ HW, BP, DP, zaujímavé projekty