



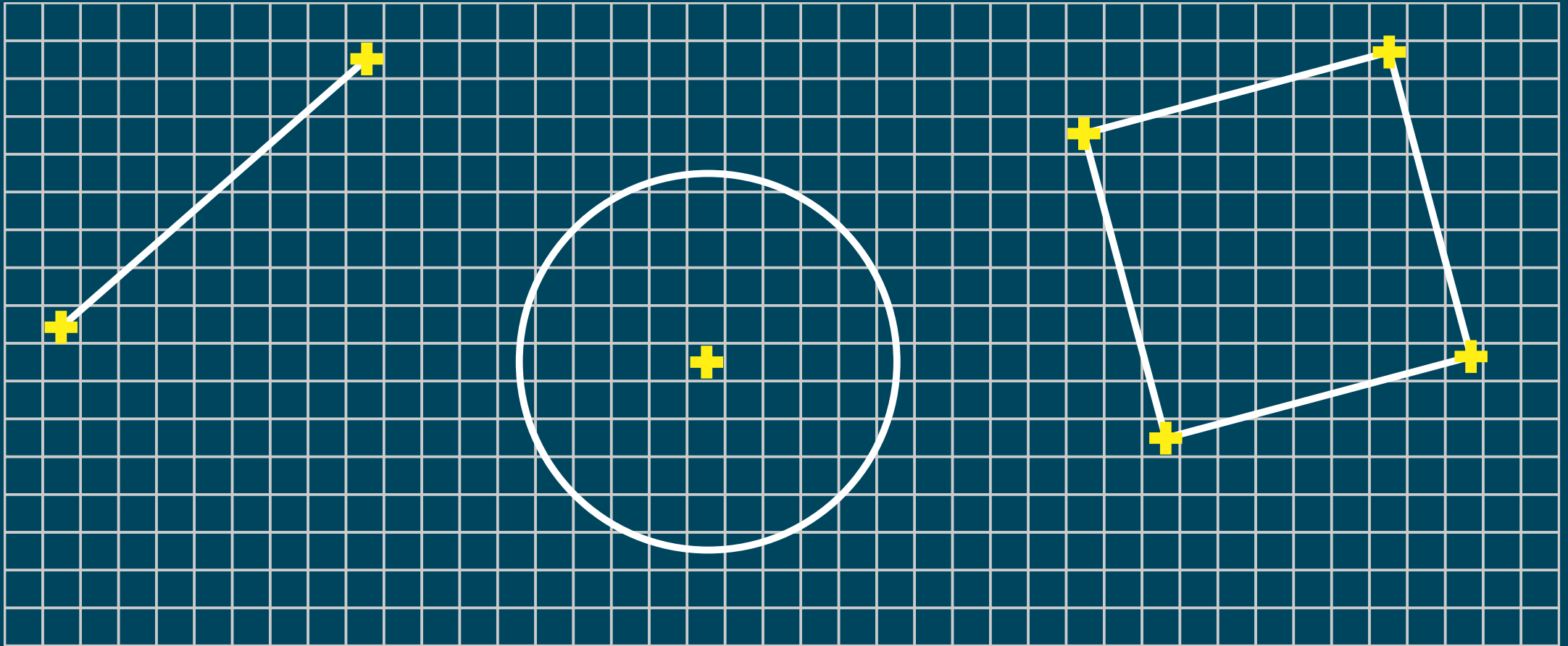
IZG

Generování základních objektů v rastru

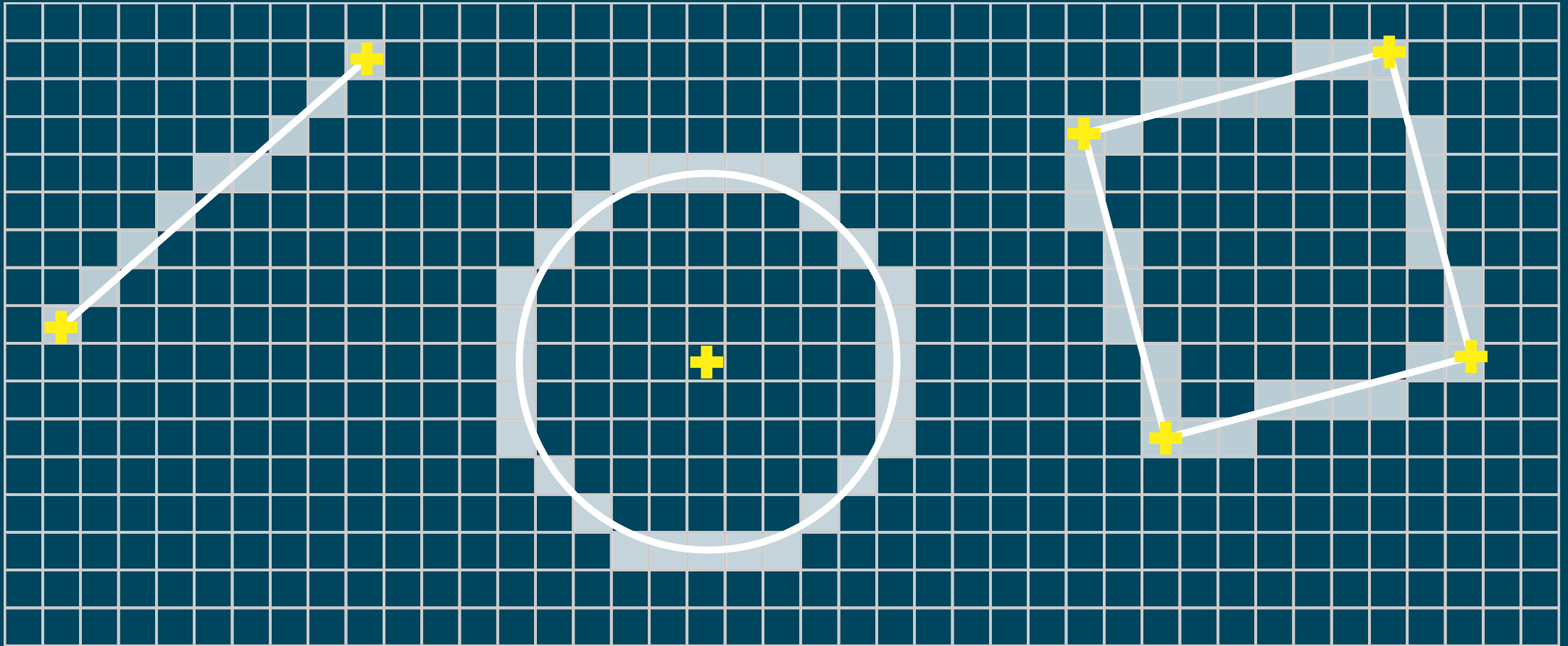
Ing. Marek Vaško

Ing. Michal Španěl Ph.D.

Rasterizácia



Rasterizácia



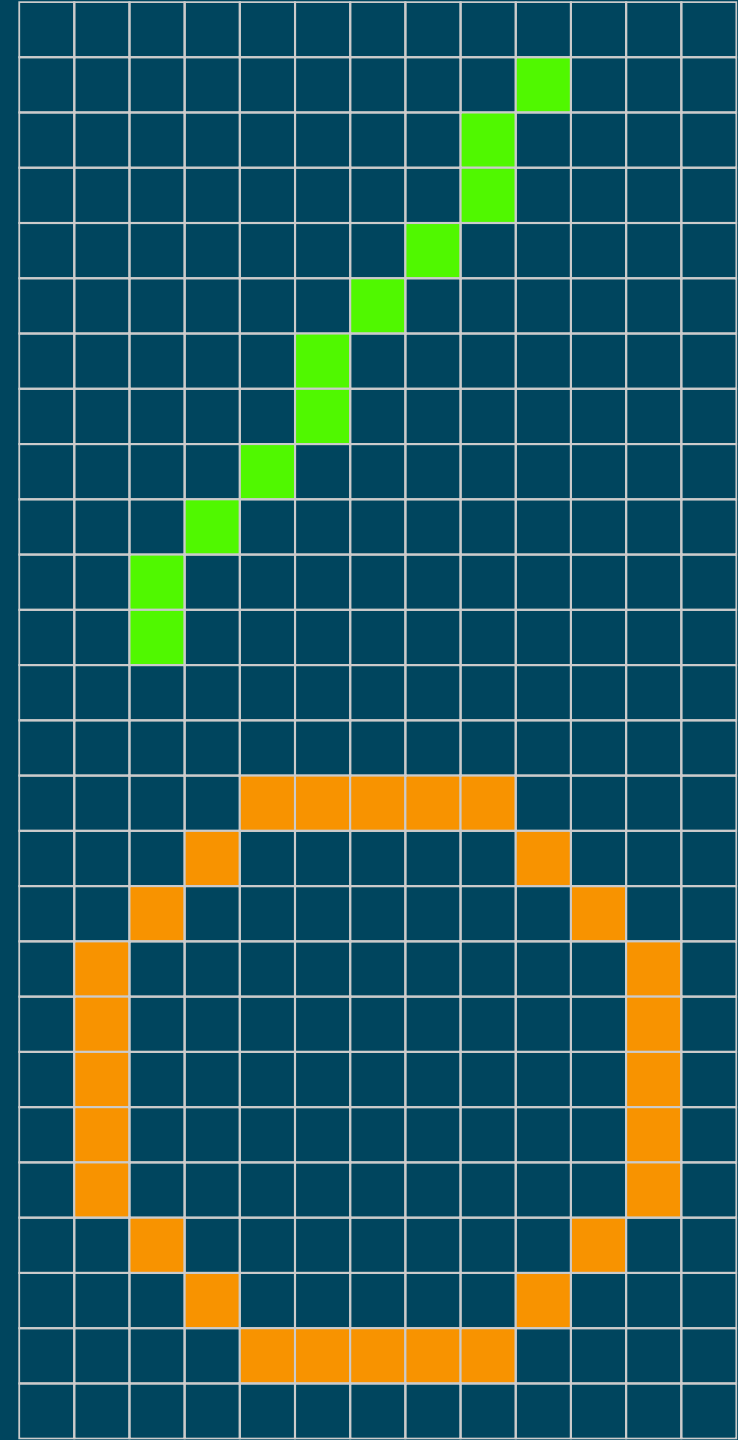
Obsah cvičenia

1. Rasterizácia úsečky

- Smernicový tvar priamky
- Algoritmy DDA, FX DDA
- Problémy rasterizačných algoritmov
- Úloha č. 1

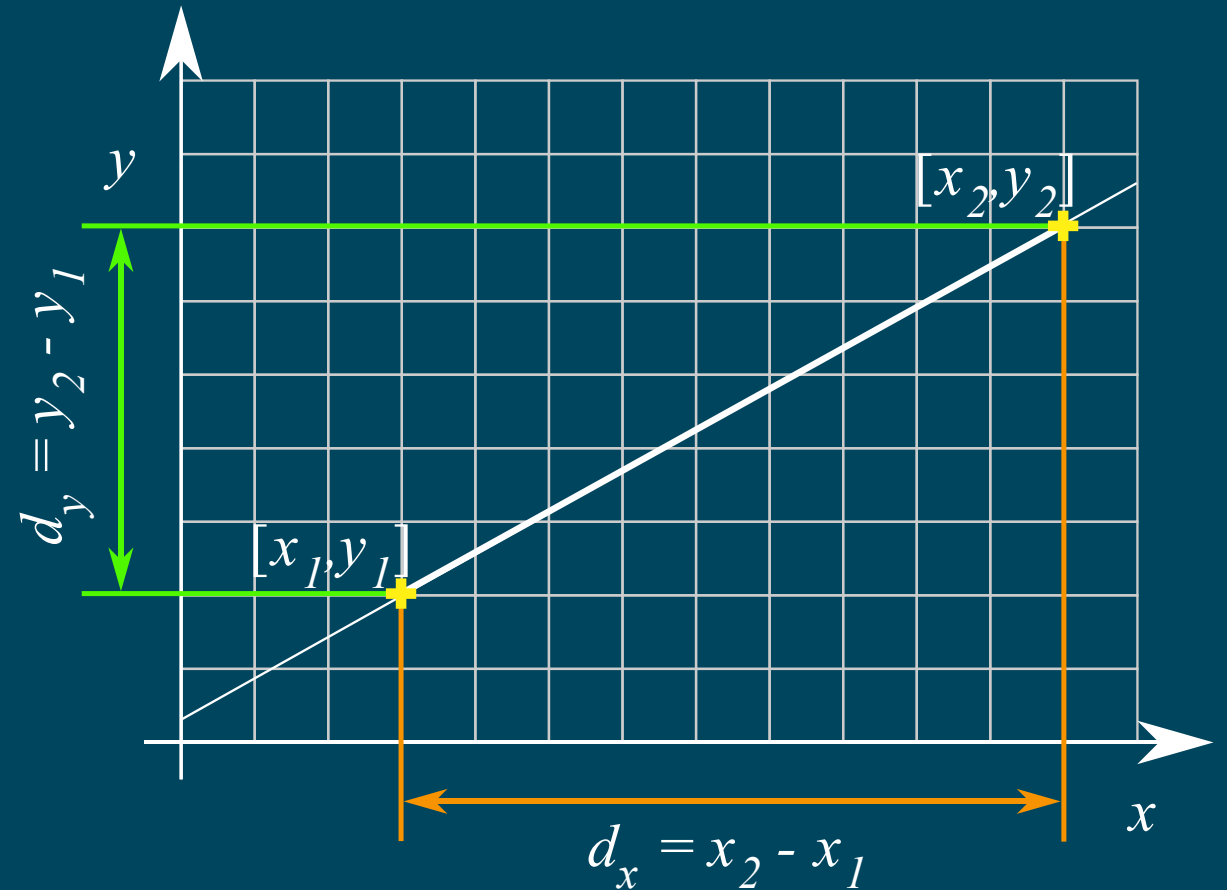
2. Rasterizácia kružnice

- Kružnica v 2D
- Algoritmus MidPoint
- Symetria kružnice
- Úloha č. 2



Smernicový tvar priamky

- $y = kx + q$,
- $k = \frac{d_y}{d_x}$
- „Iterace“ cez x

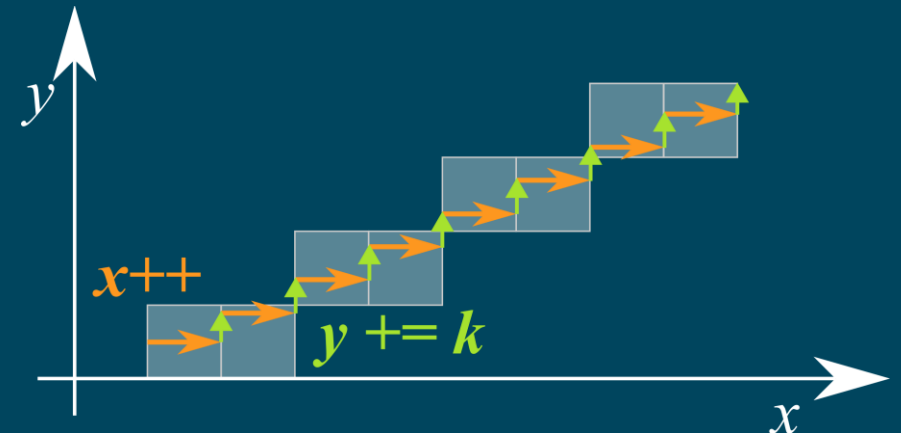


Digital Differential Analyser

DDA

- Efektívny?
- Jednoduchý
- Floating point aritmetika
- Verzia s kontrolou chyby:
 - $E \geq 0.5 \Rightarrow y := y + 1$; $E = E - 1$

```
void 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, round(y));
        y += k;
    }
}
```



Fixed-point DDA

FX DDA

- Optimalizácia DDA pre fixed point
- Jednoduchý
- Efektívny
- **FX_BITS = 8**

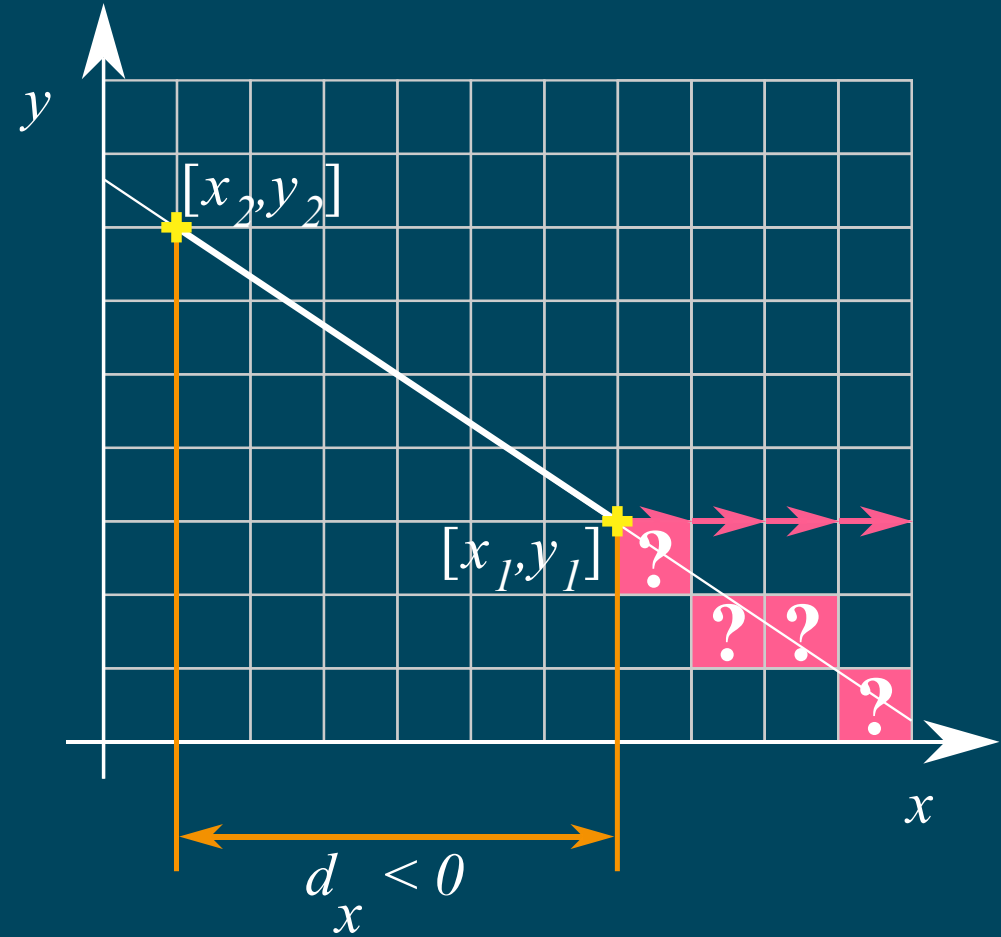
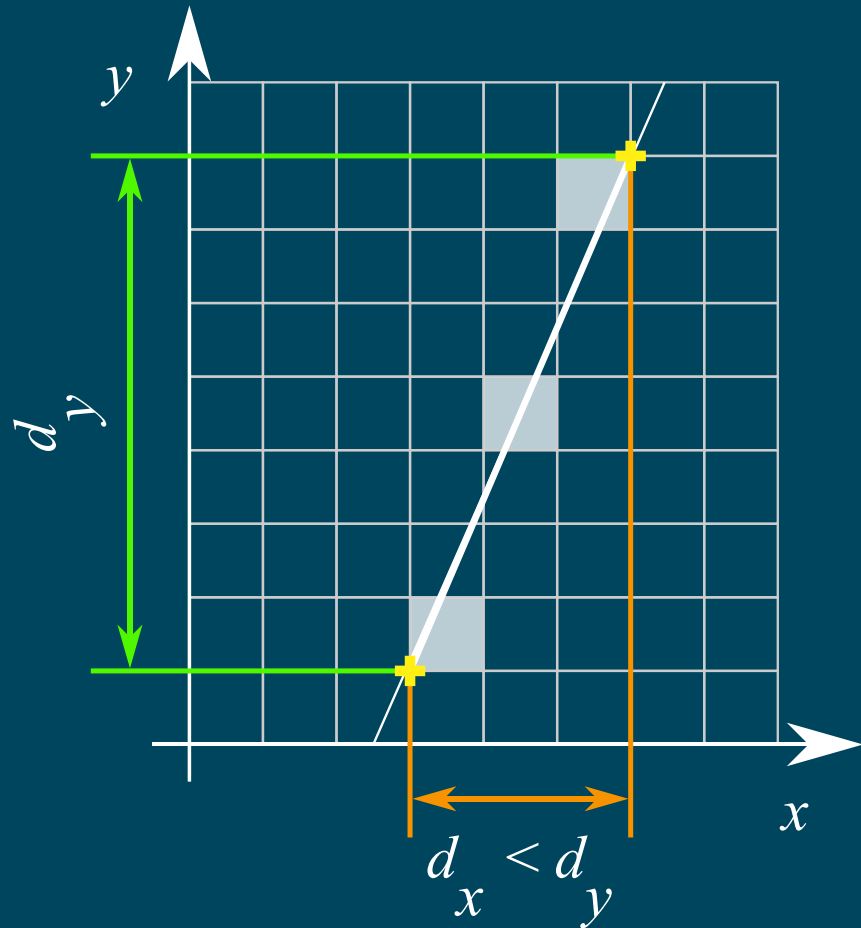
```
void LineDDAFX(int x1, int y1, int x2, int y2)
{
    int y = y1 << FX_BITS;
    const int k = ((y2-y1)<<FX_BITS)/(x2-x1);
    for (int x = x1; x <= x2; x++)
    {
        putPixel(x, y >> FX_BITS);
        y += k ;
    }
}
```

FX bits

y	0	1	1	0	1	1	1	0	1	0	1	0	1	13.83
k	0	0	0	0	0	0	1	0	1	0	0	1	1	0.32
Σ	0	1	1	1	0	0	0	1	0	1	0	0	0	14.15

y - putPixel

Problémy s DDA a FX DDA



Problémy s DDA a FX DDA

Úloha č. 1

1. ½ kvadrantu 1 a 4

- **HOTOVÉ!**

2. Rovnaký začiatočný/koncový bod

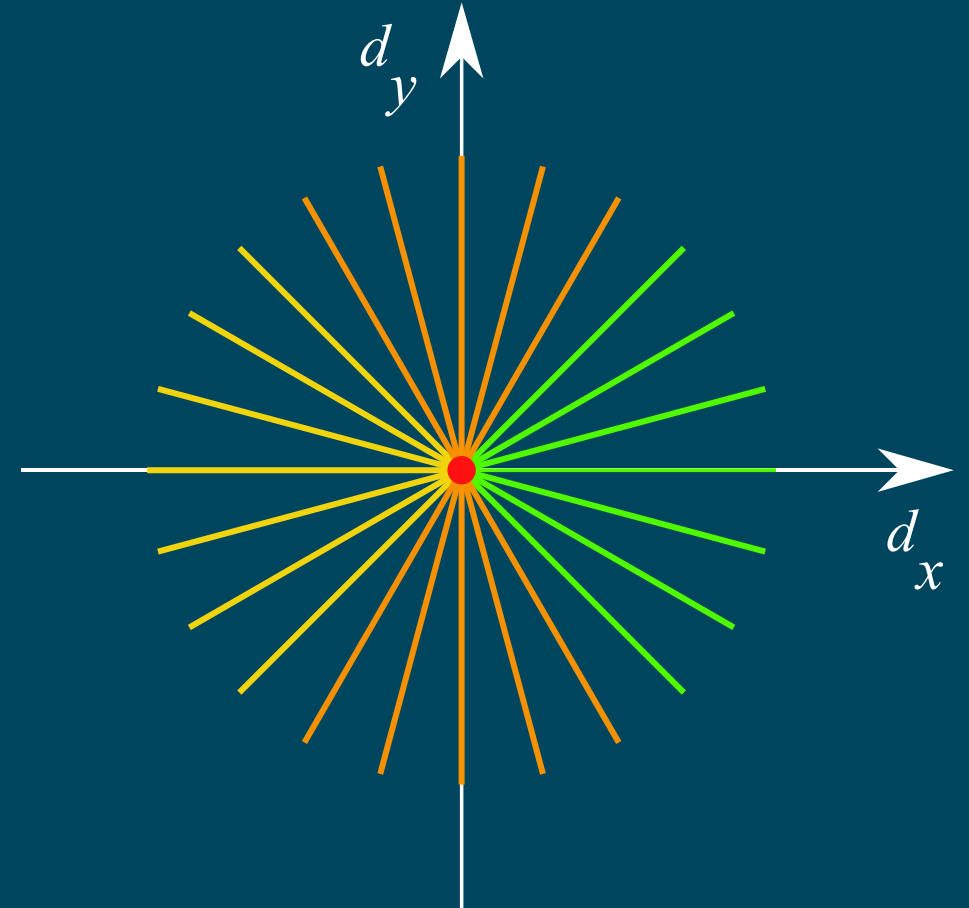
- $[x_1 \ y_1] = [x_2 \ y_2]$
- **Neplatná úsečka**

3. ½ Kvadrantov 2 a 3

- $d_x < 0$
- Výmena $[x_1 \ y_1] \Leftrightarrow [x_2 \ y_2]$

4. Horná ½ kvadrantov 1, 2, 3 a 4

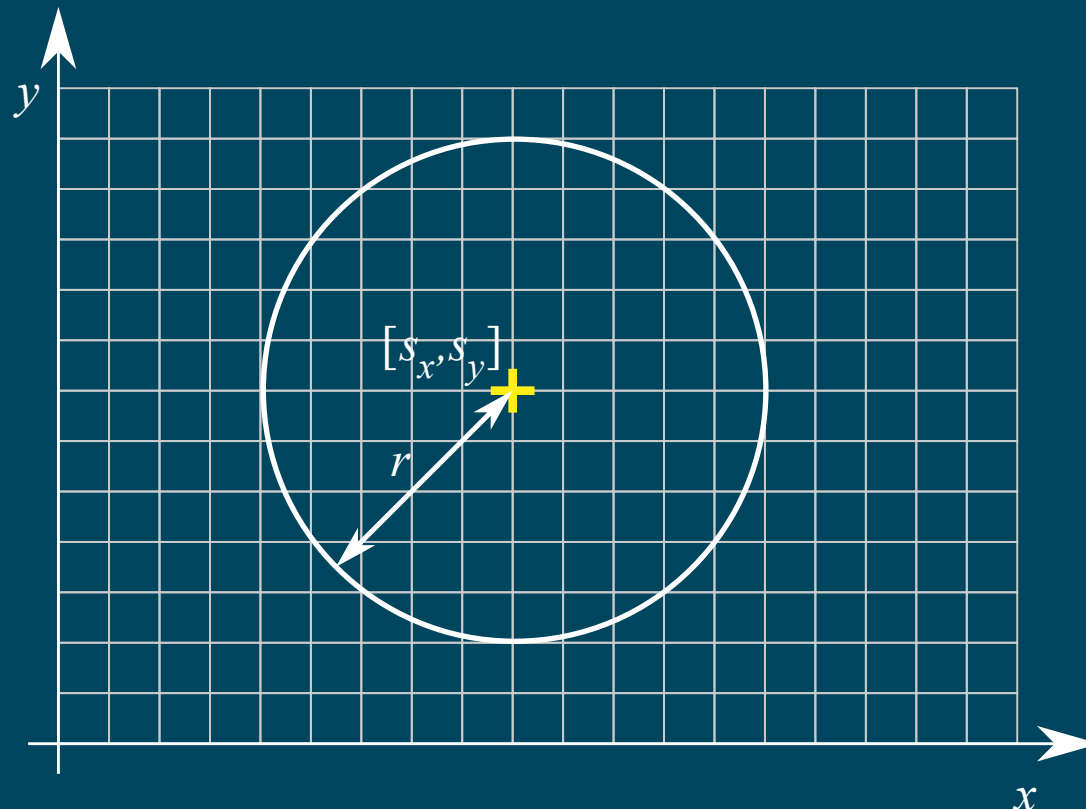
- $|d_x| < |d_y|$
- Záměna osí $X \Leftrightarrow Y$
- Záměna súradníc pri zápise



Kružnica v 2D

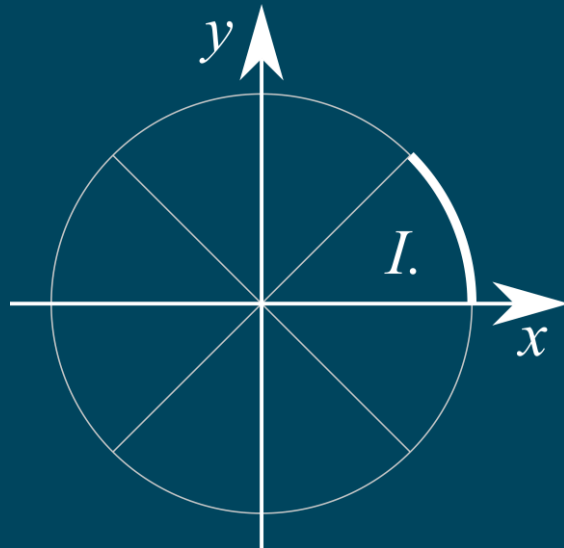
$$(x - s_x)^2 + (y - s_y)^2 = r^2$$

- $[s_x, s_y]$ – stred kružnice
- r – polomer



Algoritmus MidPoint

- Celočíselná aritmetika
- Jednoduchý
- Efektivní
- **Střed kružnice iba [0, 0]!**
- **Iba I. oktant!**



```
void circleMidPoint(int R) {  
    int x=R, y=0, P=1 - R;  
    int X2=2 - 2*R, Y2=3;  
  
    while (x >= y) {  
        put8CirclePixels(x, y);  
        if (P >= 0) {  
            P += X2; X2 += 2; x--;  
        }  
        P += Y2; Y2 += 2; y++;  
    }  
}
```

Problémy algoritmu MidPoint

Úloha č. 2

1. Posuv stredu súradnicovej sústavy

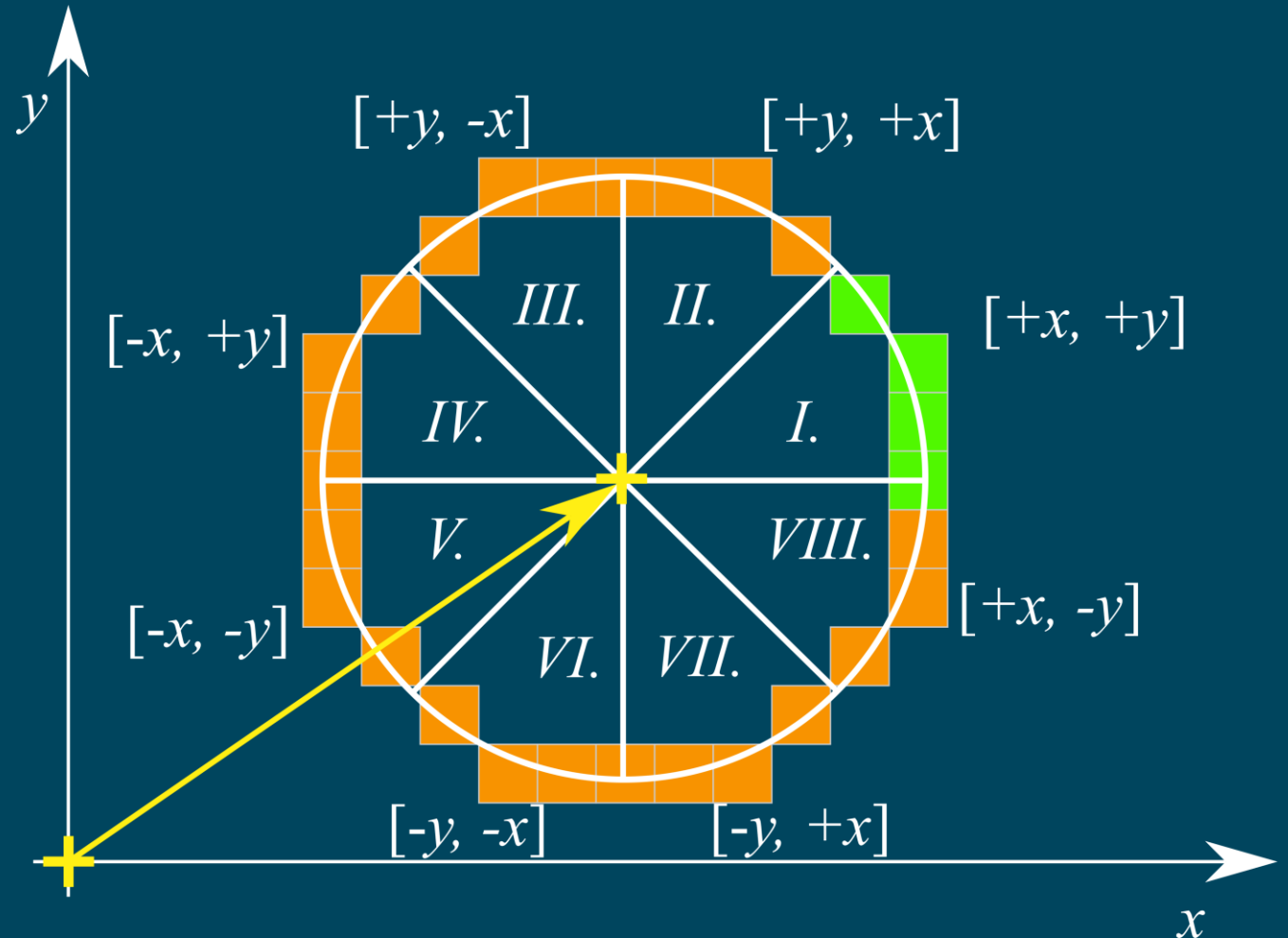
- $[0 \ 0] \rightarrow [s_x \ s_y]$

2. Oktant I.

- $s_x + x, \ s_y + y$

3. Oktant II.-VIII.

- 8-symetria
- Zvyšné kombinácie $\pm x, \pm y$
- Výmena súradníc $X \Leftrightarrow Y$



Úlohy

Využite funkcie

- `swap(..., ...)`, `abs(...)`, `setPixelColor(..., ..., ...)`

Riešenie problémov DDA

- Doplniť funkciu `rasterizeLine()`
- [2 body] – Hint: Slide 9

Symetria kruhu

- Doplniť funkciu `put8PixelsOfCircle()`
- [1 bod] – Hint: Slide 12

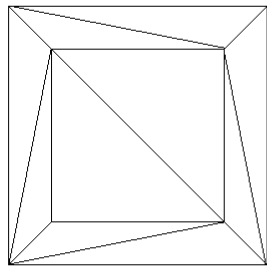
Ovládanie aplikácie:

- Kreslenie kružnice – pravé tlačítko myši + drag
- Kreslenie úsečky – ľavé tlačítko myši + drag
- Testovací vzor – klávesa T
- Vykresľovanie objektov – klávesy D, B, N, M
- Vymazanie obrazovky – klávesa C

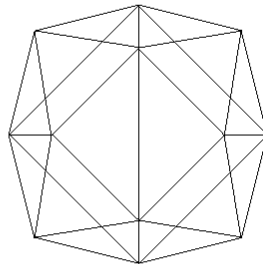
D



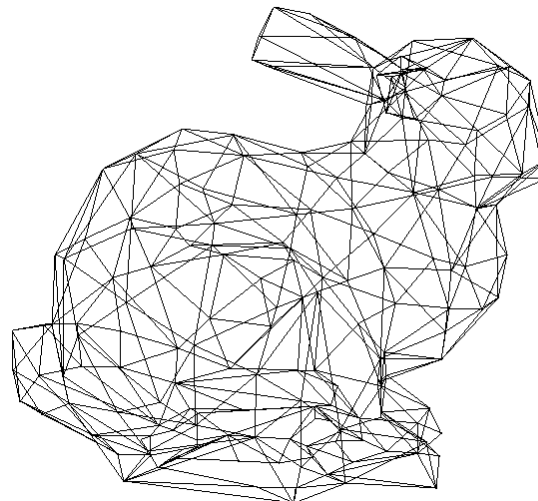
B



N



M



T

