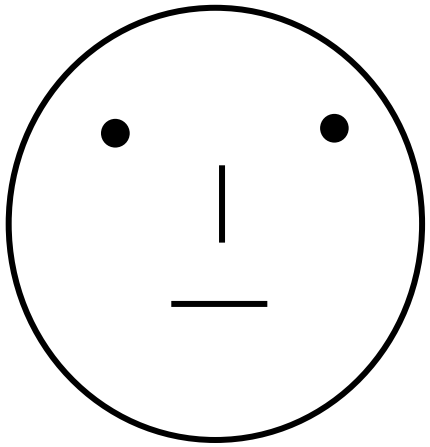
Computergrafik SS 2010
Oliver Vornberger

Kapitel 3:
2D-Grundlagen

Vorlesung vom 12.04.10

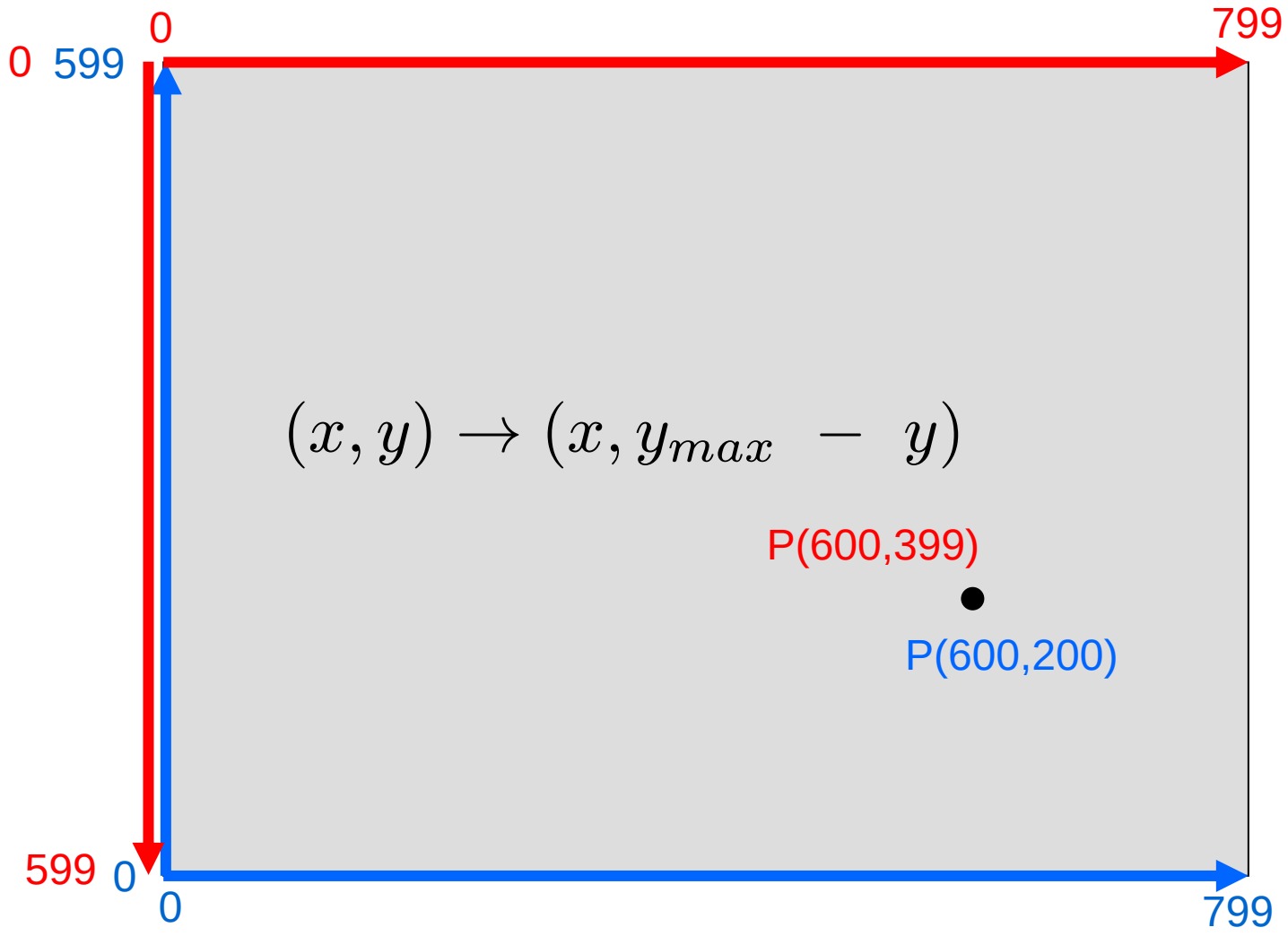
Classroomquiz

Punkt, Punkt, Komma, Strich, …

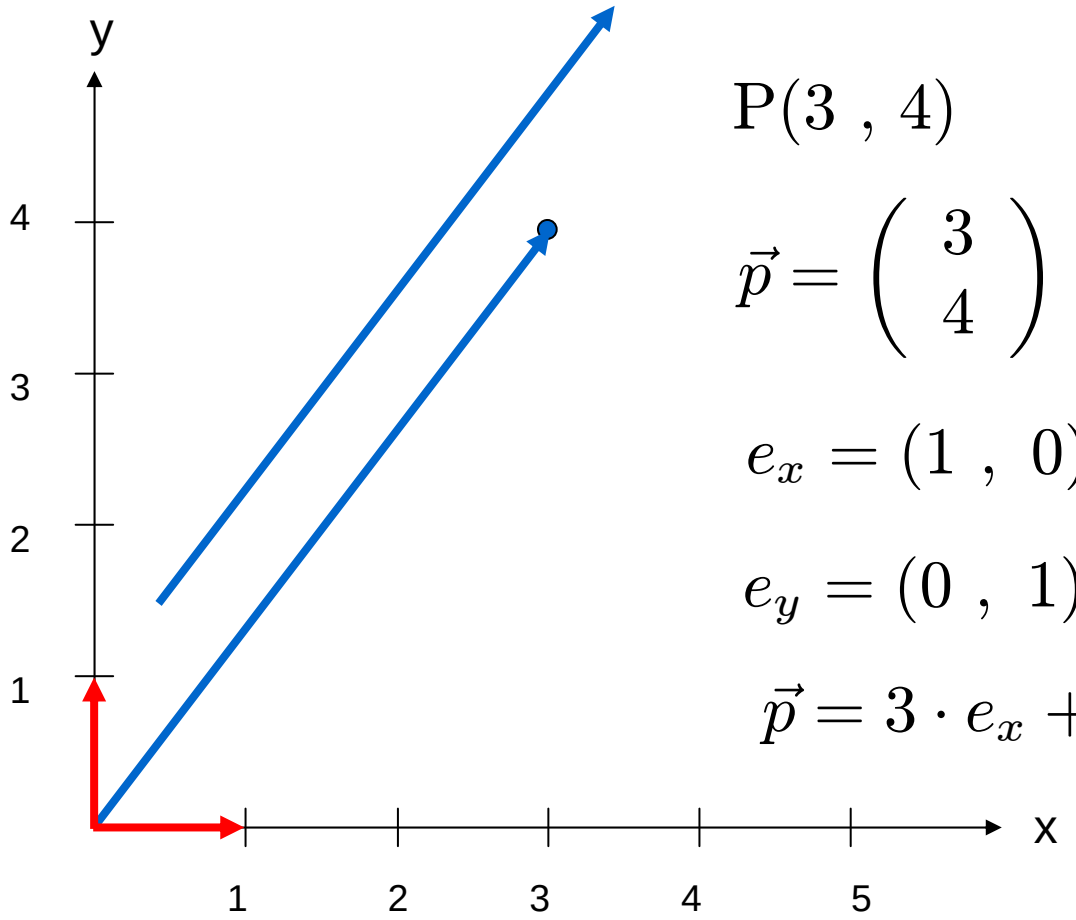


… fertig ist das Mondgesicht !

Koordinatensysteme



Punkt + Vektor



$$\mathrm{P}(3\ ,\ 4)$$

$$\vec{p} = \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (3\ ,\ 4)^T$$

$$e_x = (1\ ,\ 0)^T$$

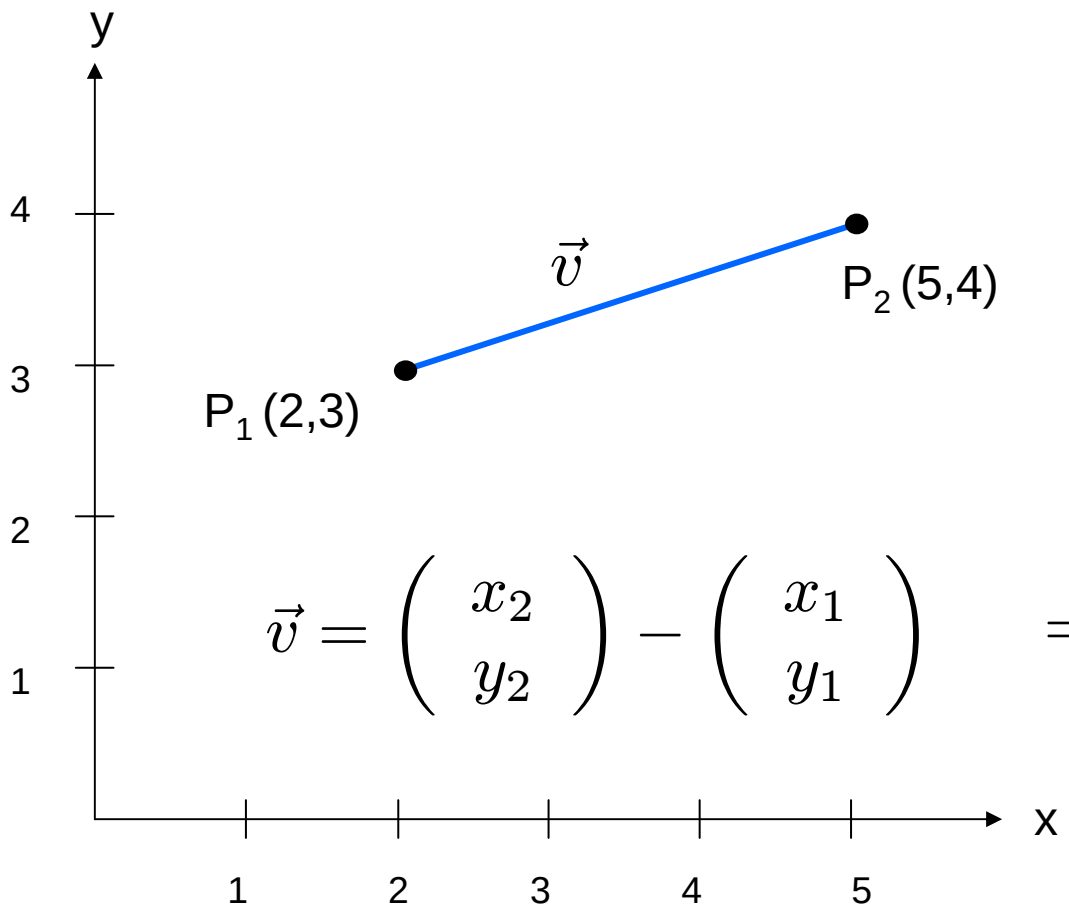
$$e_y = (0\ ,\ 1)^T$$

$$\vec{p} = 3 \cdot e_x + 4 \cdot e_y$$

setPixel(int x, int y)

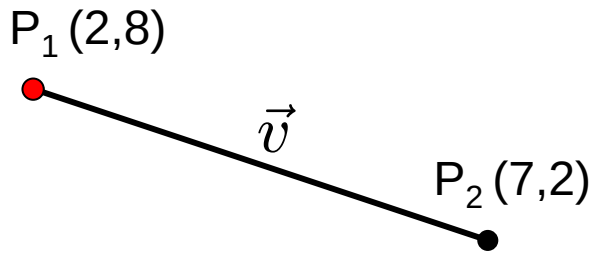
```
setPixel(3,4);

setPixel((int)(x+0.5),(int)(y+0.5));
```

Linie



$$\vec{v} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} - \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}$$

Parametrisierte Gradengleichung



$$g : \vec{u} = \vec{p_1} + r \cdot \vec{v};\ r \in \mathbb{R}$$
$$l : \vec{u} = \vec{p_1} + r \cdot \vec{v};\ r \in [0;1]$$

1.0000

$$P = (1-t) \cdot P_1 + t \cdot P_2$$

$$d = \|\overline{P_1 P_2}\| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$step = \frac{1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

VectorLine

```
int x1,y1,x2,y2,x,y,dx,dy;
double r, step;

dy = y2-y1;
dx = x2-x1;

step = 1.0/Math.sqrt(dx*dx+dy*dy);
for (r=0.0; r <= 1; r=r+step) {
  x = (int)(x1+r*dx+0.5);
  y = (int)(y1+r*dy+0.5);
  setPixel(x,y);
}
```

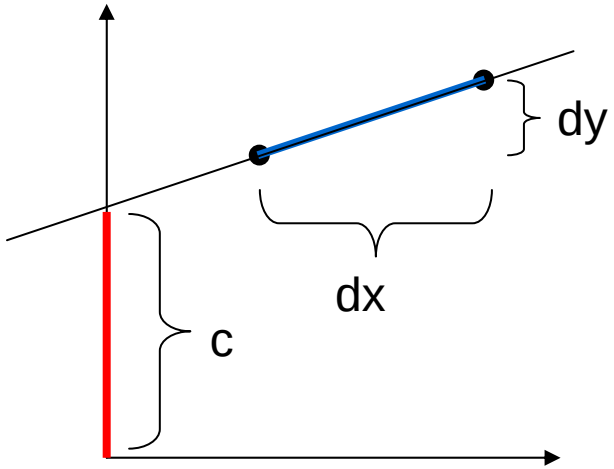
Gradengleichung als Funktion

$$y = f(x) = s \cdot x + c$$

$$s = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\frac{y_1 - c}{x_1 - 0} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$c = \frac{y_1 \cdot x_2 - y_2 \cdot x_1}{x_2 - x_1}$$



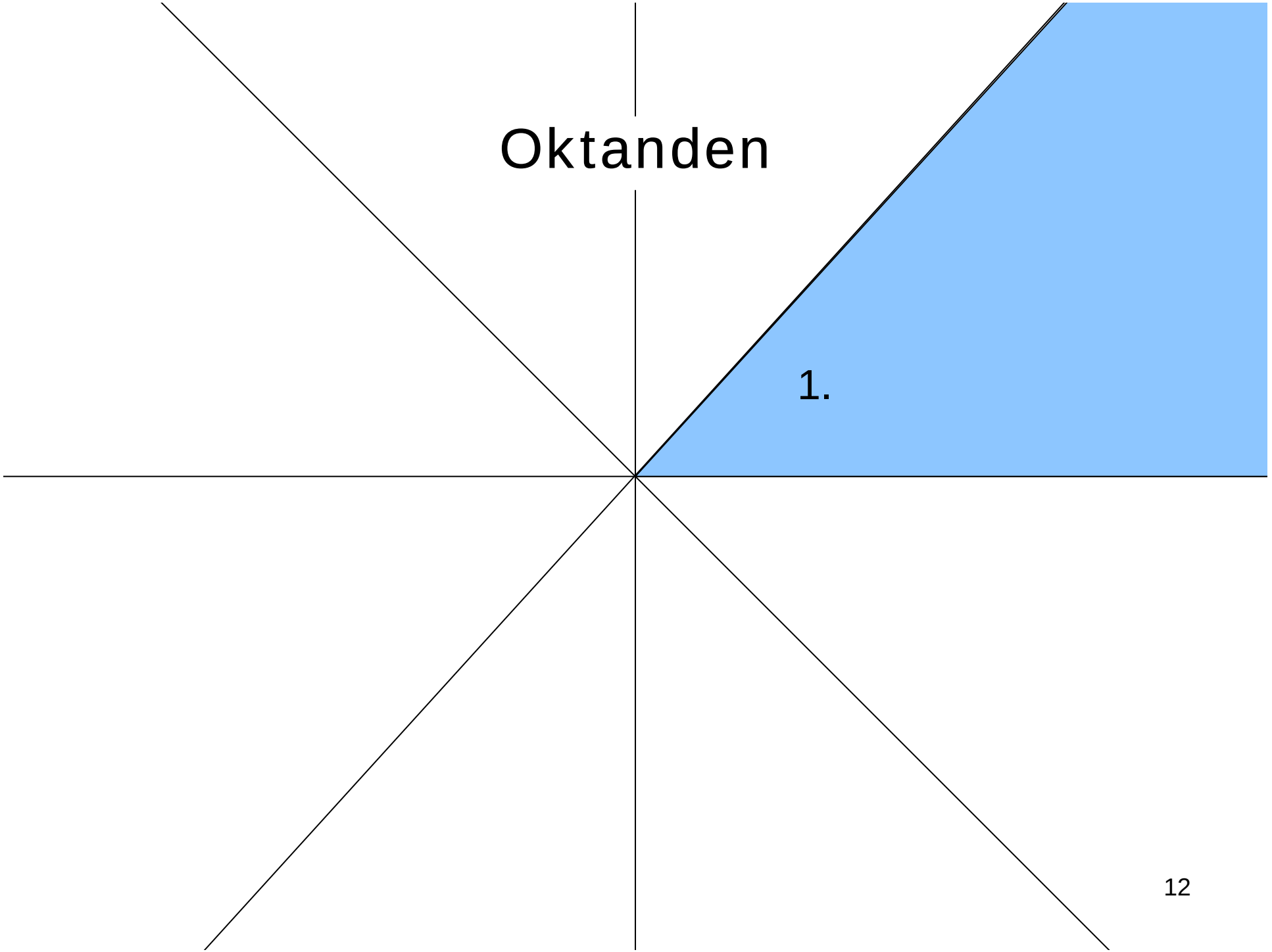
$$y = \frac{y_2 - y_1}{x_2 - x_1} \cdot x + \frac{x_2 \cdot y_1 \; - \; x_1 \cdot y_2}{x_2 - x_1}$$

StraightLine

von links nach rechts

```
s = (double)(y2-y1)/(double)(x2-x1);
c = (double)(x2*y1-x1*y2)/(double)(x2-x1);

for (x=x1; x <= x2; x++) {
  y = (int)(s*x+c+0.5);
  setPixel(x,y);
}
```

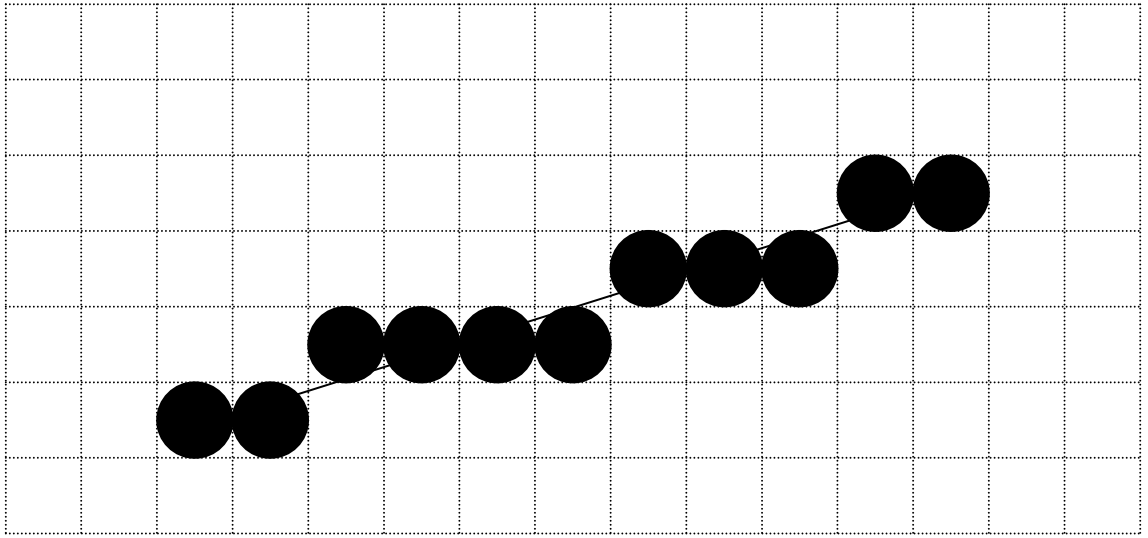
Oktanden



Bresenham

Steigung $s = \Delta y / \Delta x = 3/10 = 0.3$

Fehler $error = y_{ideal} - y_{real}$



0.0 0.3 0.6 -0.1 0.2 0.5 0.8 0.1 0.4 0.7 0.0

-0.4 -0.2 - 0.3

BresenhamLine, die 1.

```
dy = y2-y1; dx = x2-x1;
s = (double)dy/(double)dx;
error = 0.0;
x = x1;
y = y1;
while (x <= x2){
  setPixel(x,y);
  x++;
  error = error + s;
  if (error > 0.5) {
    y++;
    error = error – 1.0;
  }
}
```

Integer-Arithmetik

Mache Steigung + Fehler ganzzahlig:

$$dx := x_2 - x_1$$

$$dy := y_2 - y_1$$

$$s_{neu} = s_{alt} \cdot 2dx = \frac{dy}{dx} \cdot 2dx = 2dy$$

BresenhamLine, die 2.

```
dy = y2-y1; dx = x2-x1;
s = (double)dy/(double)dx;   delta = 2*dy
error = 0.0;
x = x1;
y = y1;
while (x <= x2){
  setPixel(x,y);
  x++;
  error = error + s;          delta
  if (error > 0.5) {          dx
    y++;
    error = error – 1.0;      2*dx
  }
}
```

multipliziere Steigung mit 2dx

Vergleich mit 0

- vergleiche **`error`** mit 0,
  d.h. verschiebe **`error`** um (x2-x1) nach unten
- verwende **`schritt`** für -2*dx

BresenhamLine, die 3.

```
dy = y2-y1; dx = x2-x1;
delta = 2*dy;
error = 0.0;                    -dx
x = x1;                         schritt= -2*dx
y = y1;
while (x <= x2){
  setPixel(x,y);
  x++;
  error = error + delta;
  if (error > dx) {             0
    y++;
    error = error - 2*dx;       + schritt
  }
}
```

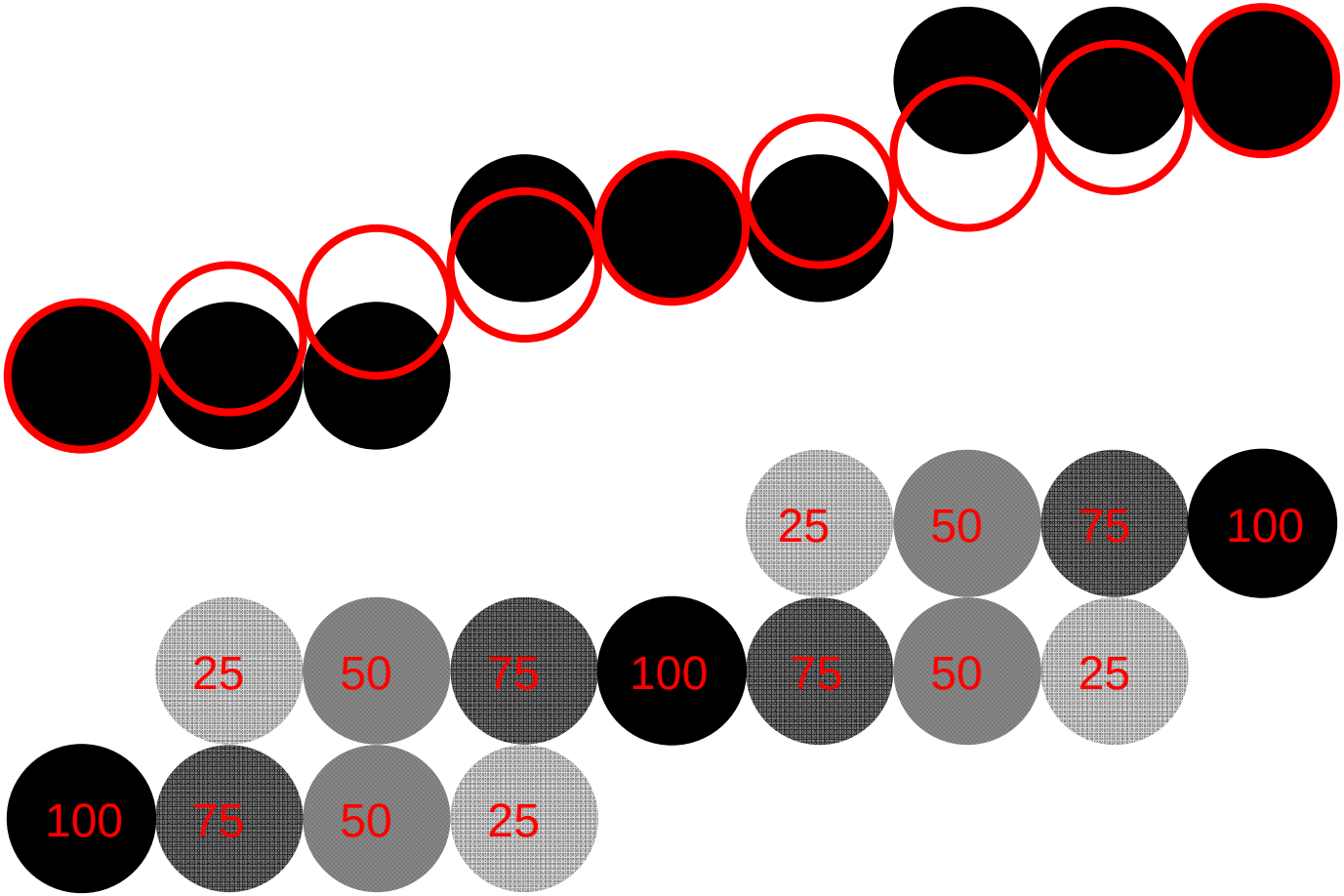
Verschiebe error um 2dx nach unten

BresenhamLine

alle 8 Oktanden durch Fallunterscheidung abhandeln:
~cg/2010/skript/Sources/drawBresenhamLine.jav.html

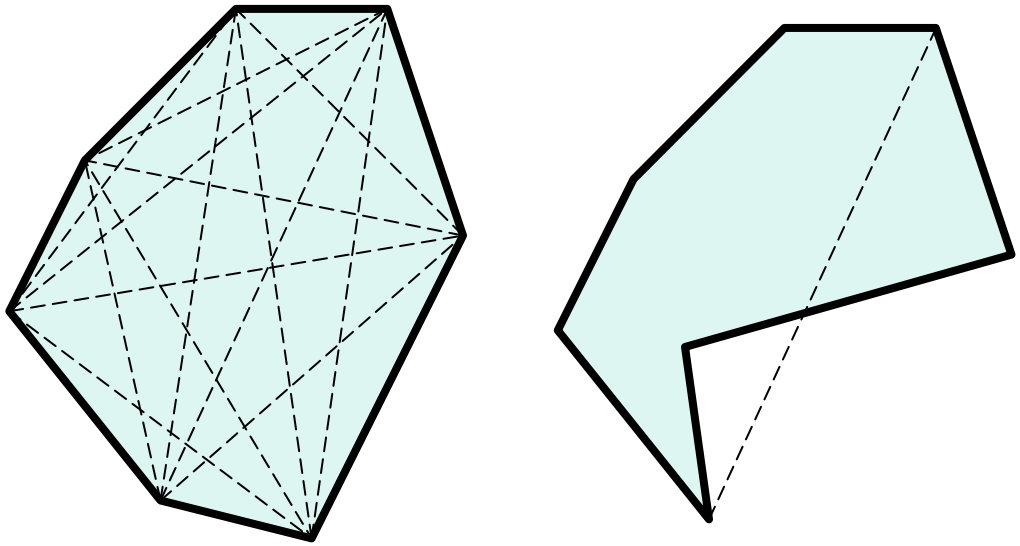
Java-Applet:
~cg/2010/skript/Applets/2D-basic/App.html

Antialiasing



Antialiasing in Adobe Photoshop

Polygon

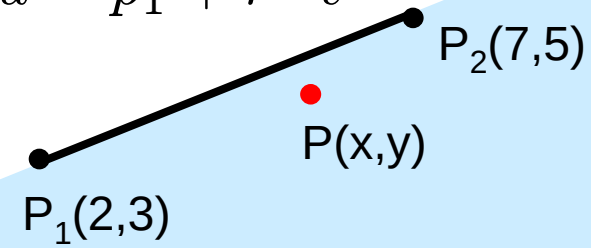


konvex konkav

Punkt versus Gerade

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} + r \cdot \begin{pmatrix} 7-2 \\ 5-3 \end{pmatrix}$$

$$\vec{u} = \vec{p_1} + r \cdot \vec{v}$$



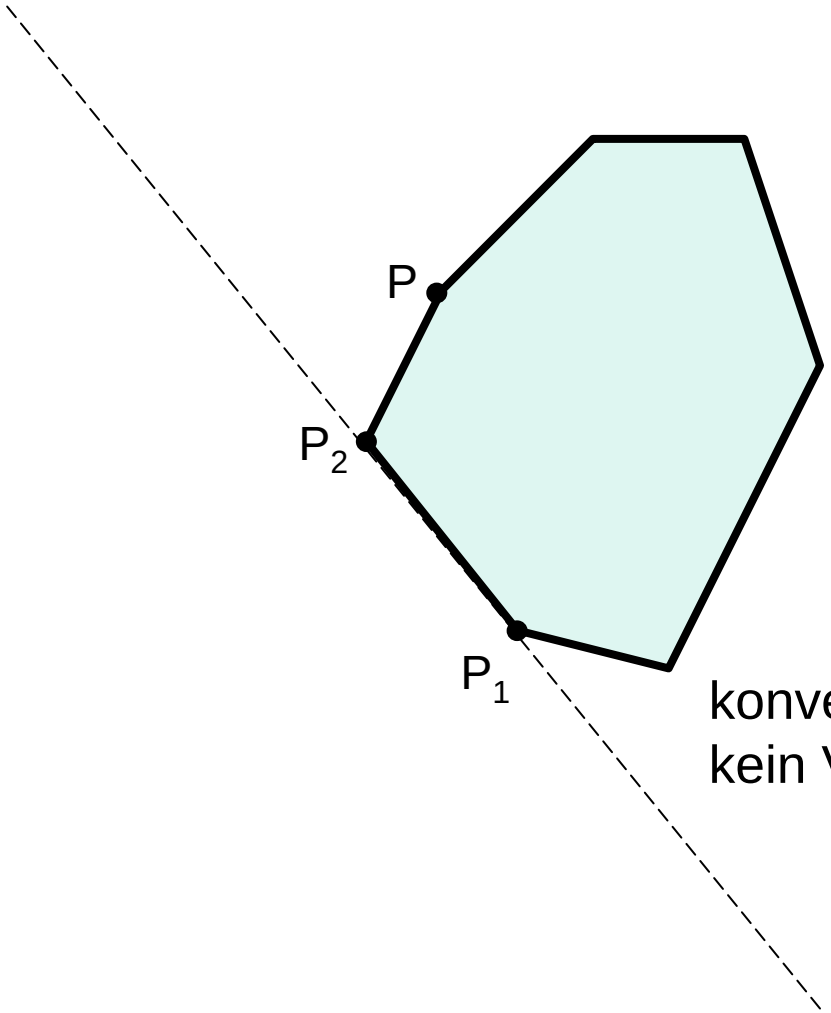
$$\begin{aligned} x &= 2 + 5r \\ y &= 3 + 2r \\ 2x &= 4 + 10r \\ -5y &= -15 - 10r \\ 2x - 5y &= -11 \\ 2x - 5y + 11 &= 0 \end{aligned}$$

$F(x,y) = 0$ falls P auf der Geraden

> 0 falls P rechts von der Geraden

< 0 falls P links von der Geraden

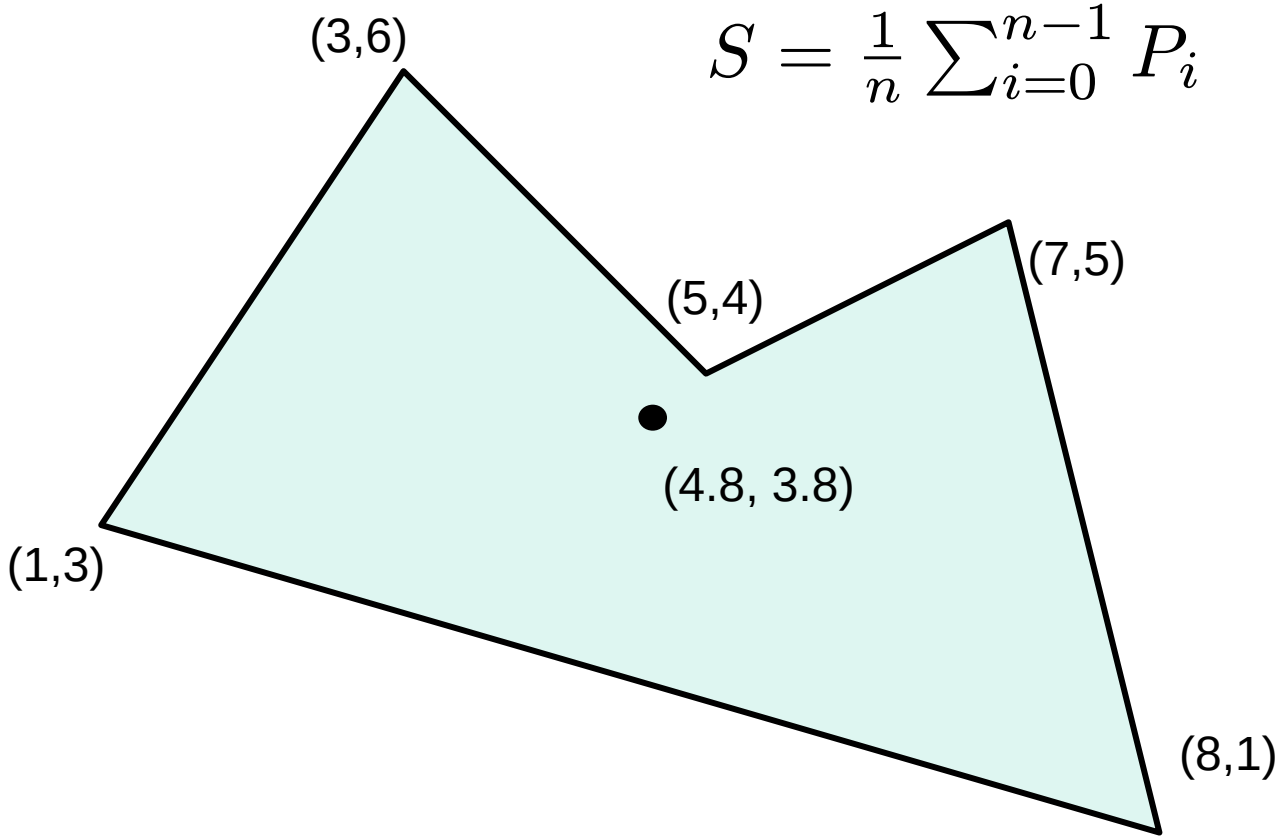
Konvexitätstest nach Paul Bourke



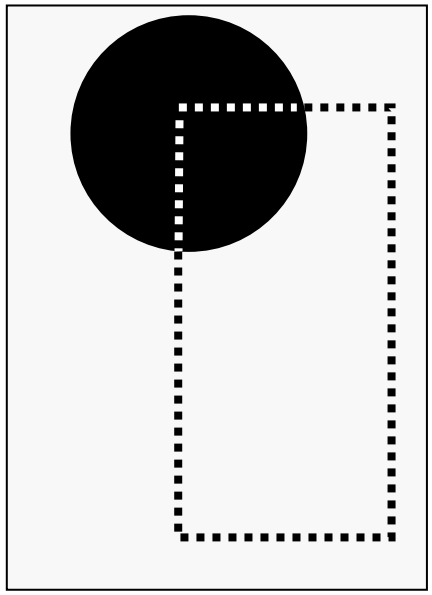
konvex, falls für alle P(x,y)
kein Vorzeichenwechsel bei F(x,y)

Schwerpunkt

$$S = \frac{1}{n} \sum_{i=0}^{n-1} P_i$$



Zeichnen und Löschen mit XOR



Pixel:	01101011
Gummiband:	11111111
XOR ergibt:	10010100
Gummiband:	11111111
XOR ergibt:	01101011

Beispiel für Gummiband:
~cg/2010/skript/Applets/2D-basic/App.html