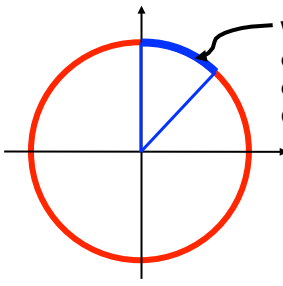
Scan-Konvertierung von Kreisen

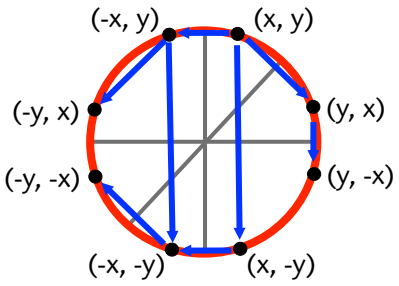
Nutze 8–Punkt-Symmetrie aus

- Berechne nur 1/8 eines Kreises mit $0 \leq x \leq \frac{r}{\sqrt{2}}$

```
drawCirclePoint(x,y):
    drawPixel(x,y)
    drawPixel(y,x)
    drawPixel(y,-x)
    :
    :
```

- Außerdem: oBdA ist Mittelpunkt = Ursprung





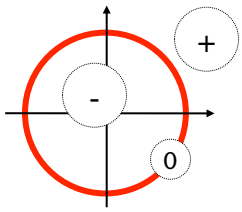
Implizite Kreisgleichungen

- Definiere $F(x, y) = x^2 + y^2 - r^2$

  Für (x,y) auf dem Kreis: F(x,y) = 0

  falls F(x,y) > 0 ⇒ (x,y) außerhalb

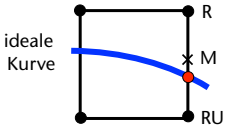
  und F(x,y) < 0 ⇒ (x,y) innerhalb

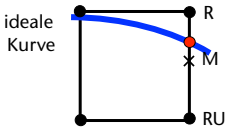


- In jedem x-Schritt: wähle R oder RU

  also: $F(M) \geq 0$ ⇒ RU

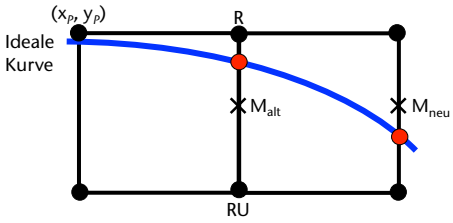
  und: $F(M) < 0$ ⇒ R





- Definiere wieder eine Entscheidungsvariable $d = F(M)$

1. Fall: $d_{alt} < 0 \rightarrow R$



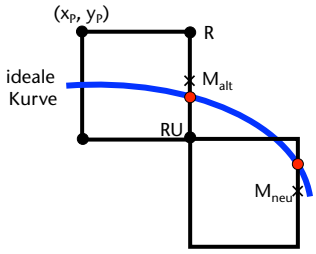
$$d_{alt} = F(x_P + 1, x_P - \tfrac{1}{2}) = (x_P + 1)^2 + (y_P - \tfrac{1}{2})^2 - r^2$$

$$d_{neu} = F(x_P + 2, x_P - \tfrac{1}{2}) = (x_P + 2)^2 + (y_P - \tfrac{1}{2})^2 - r^2$$

$$d_{neu} = d_{alt} + (2x_P + 3)$$

$$d_{\text{neu}} = d_{\text{alt}} + \Delta_R, \quad \text{mit } \Delta_R = 2x_P + 3$$

2. Fall: $d_{alt} \geq 0 \rightarrow$ RU



$$d_{neu} = F(x_P + 2, y_P - \tfrac{3}{2}) = (x_P + 2)^2 + (y_P - \tfrac{3}{2})^2 - r^2$$

$$d_{neu} = d_{alt} + (2x_P - 2y_P + 5)$$

$$d_{\text{neu}} = d_{\text{alt}} + \Delta_{RU}, \quad \text{mit } \Delta_{RU} = 2x_P - 2y_P + 5$$

- Kleines Problem: Δ_{RU} und Δ_R hängen von x_P und y_P ab!
- Idee: Bilde inkremente Differenzen 2-ter Ordnung
- Aktualisiere in beiden Fällen jeweils Δ_{RU} und Δ_R :

Fall R	Fall RU
$\Delta_R(x,y) = 2x + 3$	$\Delta_R(x,y) = 2x + 3$
$\Delta_R(x+1,y) = 2(x+1) + 3 = \Delta_R(x,y) + 2$	$\Delta_R(x+1,y-1) = 2(x+1) + 3 = \Delta_R(x,y) + 2$
$\Delta_{RU}(x,y) = 2x - 2y + 5$	$\Delta_{RU}(x,y) = 2x - 2y + 5$
$\Delta_{RU}(x+1,y) = 2(x+1) - 2y + 5 = \Delta_{RU}(x,y) + 2$	$\Delta_{RU}(x+1,y-1) = 2(x+1) - 2(y-1) + 5 = \Delta_{RU}(x,y) + 4$

- Weiteres kleines Problem:
  - Durch Startbedingung ist d (zunächst) kein Integer:
  - Sei r eine Ganzzahl. Starte bei (0, r)
  - Nächster Mittelpunkt M liegt bei (1, $r - ½$)
  - Somit: $F(1, r - 1/2) = 1 + (r^2 - r - 1/4) - r^2$
    $= 5/4 - r$
- Beobachtung: d kann nur Werte aus ..., -3/4, 1/4, 5/4, ... annehmen
- Definiere neue Entscheidungsvariable $d' := d - \frac{1}{4}$
- Es gilt: $d' < 0 \Leftrightarrow d < 0$
- Rechne also nur mit d'

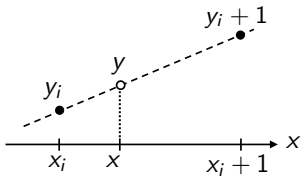
Midpoint-Algo für Kreise

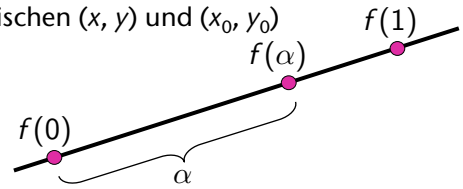
```
x ← 0, y ← r,
d ← 1-r
ΔR ← 3
ΔRU ← -2r + 5
while y > x:
   drawCirclePixel(x,y)
   x += 1
   if d < 0:
      d += ΔR
      ΔRU += 2
   else:
      d += ΔRU
      ΔRU += 4
      y -= 1
   ΔR += 2
```

Interpolation von Attributen

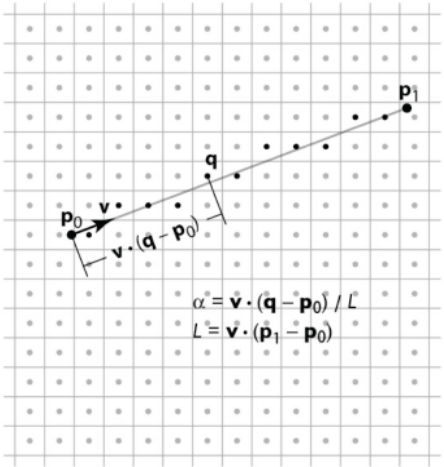
- Häufig haben Eckpunkte weitere Attribute (außer der Pos.)
  - Z.B. verschiedene Farben
- Ziel: ein gleichmäßiger Farbverlauf entlang der Linie
- Idee: lineare Interpolation
- Im 1D: $f(x) = (1 - \alpha)y_0 + \alpha y_1$

  mit $\alpha = (x - x_0)/(x_1 - x_0)$
- Im 2D ist α gerade die normierte(!) Distanz zwischen (x, y) und (x_0, y_0)





- Pixel liegen oft nicht genau auf der Linie
- Definiere 2D Funktion zur Projektion auf die Linie
  - Ist linear in x und y



- Pixel liegen oft nicht genau auf der Linie
- Definiere 2D Funktion zur Projektion auf die Linie
  - Ist linear in x und y
  - Für die Interpolation kann DDA verwendet werden