

Bresenham von (2, 1) nach (5, 5)

Oktant 2 ~~1~~ → x und y tauscht  
 $dx = 3$   $dy = 4$

X	y	e	plot
2	1	2	(2,1)
3		-6	
	2	0	(3,2)
4		-8	
	3	-2	(4,3)
	4	4	(4,4)
5		-4	
	5	2	(5,5)

```
// (x1, y1), (x2, y2) Ganzzahlig
// x1 < x2, y1 < y2
x = x1; y = y1;
dx = x2 - x1; dy = y2 - y1;
e = 2 * dy - dx; // Initialisierung
for(i = 1; i <= dx; i++){
    // Schleife fuer x
    plot(x, y);
    if (e >= 0) {
        // oberen Punkt Zeichnen (y erhoehen)
        ++y;
        e -= 2 * dx;
    }
    ++x;
    e += 2 * dy;
}
plot(x, y);
```

Notiz: mehr 1, 2, 8, 4

Oktant  
 da nur 0 oder 1  
 Änderung

## Brenham Änderungen

1)  $|dx| > |dy| \rightarrow x \text{ \& } y \text{ tauschen}$

2)  $dx < 0 \rightarrow x++ \rightarrow x--$ ,  $dx = -dx$

3)  $dy < 0 \rightarrow y++ \rightarrow y--$ ,  $dy = -dy$

## Ablanten

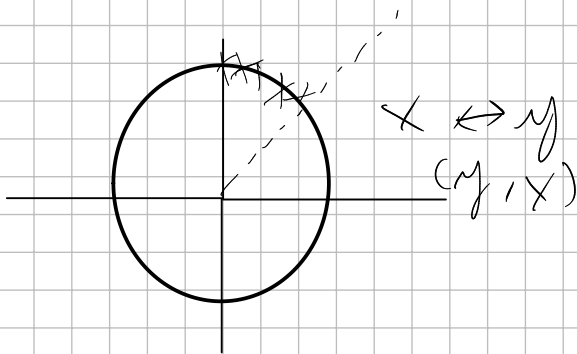
1) 2, 3, 6, 7

2) 3, 4, 5, 6

3) 5, 6, 7, 8

---

Änderung für Brenham für Oktant 1 für Kreise



Es wird Kreis aus Oktant 1 berechnet und das  
durch Korrigieren und drehen der Punkte

Problem bei der Abl von Pixel auf Texturgröße & deren Lag.

overampling  
(Pixelrate feiner als  
Texture)



Texture höher  
auflösen

underampling  
(Texture feiner als  
Pixelrate)



MIP Mapping

Texture verkleinern in  
Potenzen von 2

Wie viele mehr Speicher in Pixel verwendet mit Mapping für ein  $16 \times 16$  Texture  
mit ein Fallstudie

$16 \times 16$

$8 \times 8$

$4 \times 4$

$2 \times 2$

$1 \times 1$

256

64

16

4

1



85

## Transformationen

Die folgenden Transformationen sollen auf 3D-Objekt in der angegebenen Reihenfolge angewendet werden.

1. Skalierung der x-Komponente um Faktor 5
2. Spiegelung an y-z-Ebene
3. Rotation um  $90^\circ$  an x-Achse
4. Translation um  $(3, 2, 1)^T$

a) Transformationsmatrizen  $M_1, M_2, M_3, M_4$  in homogenen Koordinaten angeben.  
( $\sin 90^\circ = 1, \cos 90^\circ = 0$ )

$$M_1 = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_3 = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

$$M_2 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_4 = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

b) Transformieren Sie Punkt  $P$  in kartesischen Koordinaten in Punkt  $P'$ .  $P \in \mathbb{R}^3$

$$M_4 * M_3 * M_2 * M_1 P = \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix}$$

Die Transformation die als erste angewandt wird wirkt am Punkt

## Z-Buffer Alg. beschreibt

- Z-Buffer: Z-Wert der Pixel

= init:  $\infty$ , max(Z)

- Frame Buffer: init: Hintergrundfarbe

for-Schleife über alle Polygone

1) Z-Wert berechnen

2) Z-Wert < Z-Wert im Z-Buffer

• neuen Z-Wert im Z-Buffer neu

• Farbwert im Framebuffer

Polygone nicht geordnet  $\rightarrow$  Z-Buffer nicht geeignet für Transparenz

was eignet sich der Z-Buffer-Alg. nicht für Szen mit Transparenz

Zielerhebung von Eichtkorrekturverfahren

Bestimme man  $S_{\text{eff}}$  & man misst  $s_{\text{eff}}$  in

Zielerhebung von Culling-Verfahren

- Zeiterparierung bei der Schnittpunktberechnung

→ Bestimmung der Offenheit nicht mehr Teil der Szene

Projektion zur geringen Verproben

Warum ist Brennen schneller als PDR

↑  
geringer

↑  
geringer

geringer geht viel schneller