Der Algorithmus von Bresenham

Das Bresenham-Verfahren beruht im wesentlichen auf zwei grundsätzliche

Beobachtungen:

- Es reicht ein Verfahren aus um Geraden mit einer Steigung im Bereich von null bis eins darzustellen.

- Es kommen für die Linie prinzipiell immer nur zwei Punkte in Frage, die als nächstes gezeichnet werden dürfen.

Die erste Behauptung läßt sich einfach erklären. Wenn eine Gerade eine Steigung von minimal null und maximal eins hat, dann liegt sie zwischen

einer Waagerechten und einer Geraden, die einen Winkel von 45 Grad mit der X-Achse einschließt.

Es gibt natürlich auch Geraden mit einer steileren Steigung als eins. Doch alle diese Geraden kann man auch erhalten, indem man eine Gerade mir der Steigung null bis eins um die Winkelhalbierende spiegelt. Dies kann man leicht erreichen, indem man die X- und Y-Koordinaten austauscht.

Bleiben noch Geraden mit einer negativen Steigung, also "fallende" Geraden, übrig. Doch auch diese lassen sich herleiten, indem man die entsprechende Gerade an der X-Achse spiegelt. Das erreicht man durch das Umdrehen des Vorzeichens der Y-Koordinate.

Die zweite wichtige Voraussetzung des Algorithmus basiert nun auf der erstgenannten. Sie besagt, daß bei allen Geraden die aufwendigen Berechnungen unter Einbeziehung der Steigung überflüssig sind. Wenn man vom Anfangspunkt einer "Grund-Geraden" ausgeht, kommen generell nur zwei Punkte in Frage, die als nächste gezeichnet werden dürfen.

Beginnend mit dem Anfangspunkt wird kontinuierlich entschieden, ob der rechts davon liegende Punkt A oder B dargestellt werden muß. Von diesem Punkt wird wieder weiter entschieden.

Jetzt stellt sich die Frage wie entschieden wird?

Dazu muß herausgefunden werden, welcher Punkt, A oder B, näher der tatsächlichen Gerade liegt. Es wird von der Geradengleichung $y = kx + d$ ausgegangen. Bei der Bresenham-Methode wird der Einfachheit halber davon ausgegangen, daß der Anfangspunkt der Gerade durch den Ursprung geht.

Daher wird d zu null und die Gleichung vereinfacht sich zu: $y = kx$

Der Bresenham-Algorithmus berechnet die Koordinaten jedes einzelnen Punktes, indem vom zuletzt gezeichneten Punkt ausgegangen wird. Der Punkt P

besitzt die Koordinaten X und Y. Als nächster Punkt kommt entweder A oder B mit den

Koordinaten X_{+1} und Y_{+1} bzw. X_{+1} und Y.

Der Punkt A liegt oberhalb der tatsächlichen Geraden. Die tatsächliche Y-Koordinate an der Stelle X ergibt sich aus der Geradengleichung:

y = kx Daher beträgt der Abstand des Punktes A zu dieser Koordinate

a = y + 1 - kx

Ähnlich läßt sich auch b berechnen. Der Punkt B liegt unterhalb des exakten Punktes der Gerade.
b = kx - y

Jetzt kann leicht entschieden werden, welcher Punkt gezeichnet wird.

Ist a kleiner b wird A gezeichnet. Ist b kleiner a wird B gezeichnet.

Es ist also wichtig die Differenz zu berechnen:

b - a = kx - y - (y + 1 - kx)

Die Steigung kann man aus dem Quotienten der Differenzen der Koordinaten berechnen.

$$k = \frac{Y2 - Y1}{X2 - X1} = \frac{DY}{DX}$$

Diesen Term setzt man nun in den Ausdruck (b-a) ein:

$$b - a = \frac{DY}{DX} x - y - (y + 1 - \frac{DY}{DX} x)$$

Diesen Ausdruck kann man vereinfachen:

DY

Klammer auflösen -> $b - a = 2 \frac{DY}{DX} x - 2y - 1$

DX

---> (b - a) DX = 2 (DY x - y DX) - DX

Der Trick beim Bresenham-Algorithmus liegt nun darin, daß man die Differenz (b - a) nicht jedesmal völlig neu berechnet, sondern jeden neuen Punkt aus der Differenz des letzten Punktes ableitet. Für den Ausdruck (a - b) ergibt sich an der Stelle Xi und Yi:

(bi - ai) DX = 2 (DY xi - yi DX) - DX

Für den nächsten Punkt ergibt sich analog dazu:

$(bi_{+1} - ai_{+1})$ DX = 2 (DY xi_{+1} - yi_{+1} DX) - DX

Um festzustellen, wie man anhand von (bi - ai) DX den Ausdruck $(bi_{+1} - ai_{+1})$ DX berechnen kann, zieht man beide Terme voneinander ab:

$(bi_{+1} - ai_{+1})$ DX - (bi - ai) DX = 2 (DY xi_{+1} - yi_{+1} DX) - DX -

(2 (DY xi - yi DX) - DX)

Durch Umformung erreicht man daraus:

2 (DY xi_{+1} - yi_{+1} DX) - DX -2 (DY xi - yi DX) + DX

= 2 (DY xi_{+1} - yi_{+1} DX - (DY xi - yi DX)

= 2 (DY xi_{+1} - yi_{+1} DX - DY xi + yi DX)

= 2 (DY (xi_{+1} - xi) - DX(yi_{+1} - yi))

Da die X-Koordinaten immer nebeneinander liegen, gilt:

xi_{+1} - xi = 1

Jetzt setzt man Eins in den obigen Term ein:

2 (DY - DX (y_{i+1} - yi))

Entweder beträgt der Ausdruck null oder eins. Ein Wert von null tritt dann auf wenn Punkt B gesetzt wurde. In diesem Fall ist y_{i+1} - yi = 0 => 2 DY

Ein Wert von eins tritt dann auf wenn A gesetzt wurde. Es gilt y_{i+1} - yi = 1

=> 2 (DY - DX)

Die Differenz der beiden Alternativpunkte entweder 2 DY oder 2 DY - DX stellen Konstanten dar, die schon am Beginn des Programms einmal berechnet werden können, ebenfalls kann berechnet werden, welcher Alternativpunkt gesetzt werden muß.

Das Programm in Pascal

```
Procedure Line (X1, Y1, X2, Y2: word; Farbe: byte);
Var
DX, DY, DAB : Integer;
IncA, IncB, IncY :Integer;
X, Y : Integer;

Begin
If X2<X1 Then
Begin
Tausche (X1, X2); (* Die Koordinaten müssen ver- *)
Tausche (Y1, Y2); (* tauscht werden. *)
```

```
End;
If (Y1 < Y2) (* Steigung positiv ? *)
Then
IncY:=1 (* Y muß in der Schleife erhöht *)
Else (* werden *)
IncY:=-1; (* wenn nicht, Y abziehen *)
DX := X2-X1; (* Berechnung der Konstanten *)
DY := Y2-Y1; (* DX, DY, Differenz (a-b) *)
DAB := (DY SHL 1) - DX;
IncA := (DY - DX) SHL 1; (* Alternativwert für A und B *)
IncB := DY SHL 1;
X := X1;
Y := Y1;
Putpixel (X, Y, 2); (* Setze ersten Punkt *)
For X:= X1+1 To X2 Do
Begin
If DAB < 0 Then (* Es wurde zuletzt A gesetzt *)
Begin
Dab := Dab + IncB; (* Differenz ändern und *)
Putpixel (X, Y, 2); (* Punkt setzen *)
End
Else (* Es wurde zuletzt B gesetzt *)
Begin
Y := Y + IncY; (* Neue Koordinaten und *)
Dab := Dab + IncA; (* Differenz ändern *)
End;
End;
End;
```