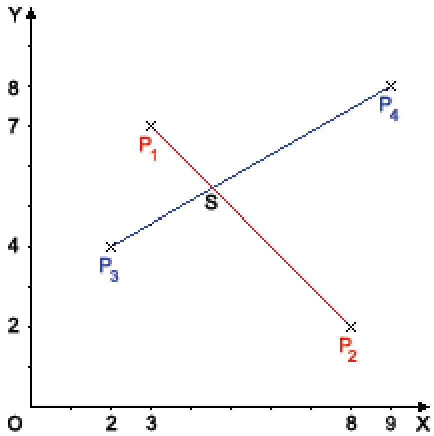
Linienschnittpunk

Folgendes Tutorial zeigt, wie mit Hilfe der Vektorrechnung, der Schnittpunkt zweier Linien gefunden werden kann. Es ist für die Programmiersprache BlitzMax entwickelt wurden und der Autor ist Oliver `Vertex' Skawronek.

Theorie



Dabei gilt für jeden Punkt auf der Strecke $\overline{P_1P_2}$ die Geradengleichung

$$a : x = \overrightarrow{OP_1} + t_a\left(\overrightarrow{OP_2} - \overrightarrow{OP_1}\right) \quad t_a \in R \cap [0;1]$$

sowie für $\overline{P_3P_4}$ die Geradengleichung

$$b : x = \overrightarrow{OP_3} + t_b\left(\overrightarrow{OP_4} - \overrightarrow{OP_3}\right) \quad t_b \in R \cap [0;1]$$

. Es gilt nun entsprechend für die Skalars t_a oder t_b einen Wert zu finden, so dass

$$\overrightarrow{OP_1} + t_a\left(\overrightarrow{OP_2} - \overrightarrow{OP_1}\right) = \overrightarrow{OP_3} + t_b\left(\overrightarrow{OP_4} - \overrightarrow{OP_3}\right)$$

gilt. Umgestellt nach t_a und t_b:

$$\begin{aligned}
\overrightarrow{OP_1} + t_a\left(\overrightarrow{OP_2} - \overrightarrow{OP_1}\right) &= \overrightarrow{OP_3} + t_b\left(\overrightarrow{OP_4} - \overrightarrow{OP_3}\right) \quad | \quad -\overrightarrow{OP_1} \\
t_a\left(\vec{OP_2} - \vec{OP_1}\right) &= \vec{OP_3} + t_b\left(\vec{OP_4} - \vec{OP_3}\right) - \vec{OP_1} \quad | \quad -t_b\left(\vec{OP_4} - \vec{OP_3}\right) \\
t_a\left(\overrightarrow{OP_2} - \overrightarrow{OP_1}\right) - t_b\left(\overrightarrow{OP_4} - \overrightarrow{OP_3}\right) &= \overrightarrow{OP_3} - \overrightarrow{OP_1}
\end{aligned}$$

Umgewandelt von der Vektorgleichung in eine Komponentengleichung ergibt das ein Gleichungssystem mit 2 Gleichungen sowie 2 Unbekannten.

$$\left| \begin{array}{l} t_a\left(\overrightarrow{OP_2}_x - \overrightarrow{OP_1}_x\right) - t_b\left(\overrightarrow{OP_4}_x - \overrightarrow{OP_3}_x\right) = \overrightarrow{OP_3}_x - \overrightarrow{OP_1}_x \\ t_a\left(\overrightarrow{OP_2}_y - \overrightarrow{OP_1}_y\right) - t_b\left(\overrightarrow{OP_4}_y - \overrightarrow{OP_3}_y\right) = \overrightarrow{OP_3}_y - \overrightarrow{OP_1}_y \end{array} \right|$$

Hier benutzen wir ganz normal das Additionsverfahren um t_a auszuschließen, so, dass t_b übrig bleibt. Dazu müssen wir die erste Gleichung so multiplizieren, dass addiert mit der zweiten Gleichung, t_a wegfällt. Damit das gelinkt, wird der Faktor f aus den Koeffizienten von t_a aus der ersten und zweiten Gleichung berechnet:

$$f = -\frac{\overrightarrow{OP_2}_y - \overrightarrow{OP_1}_y}{\overrightarrow{OP_2}_x - \overrightarrow{OP_1}_x}$$

Ausmultipliziert ergibt dies:

$$\begin{aligned} & -t_b\left(\overrightarrow{OP_4}_x - \overrightarrow{OP_3}_x\right) f - t_b\left(\overrightarrow{OP_4}_y - \overrightarrow{OP_3}_y\right) \\ & = f\left(\overrightarrow{OP_3}_x - \overrightarrow{OP_1}_x\right) + \overrightarrow{OP_3}_y - \overrightarrow{OP_1}_y \end{aligned}$$

Nach t_b umgestellt:

$$\begin{aligned} & -t_b\left(\overrightarrow{OP_4}_x - \overrightarrow{OP_3}_x\right) f - t_b\left(\overrightarrow{OP_4}_y - \overrightarrow{OP_3}_y\right) \\ & = f\left(\overrightarrow{OP_3}_x - \overrightarrow{OP_1}_x\right) + \overrightarrow{OP_3}_y - \overrightarrow{OP_1}_y \\ & t_b\left(-f\,\overrightarrow{OP_4}_x + f\,\overrightarrow{OP_3}_x - \left(\overrightarrow{OP_4}_y - \overrightarrow{OP_3}_y\right)\right) \\ & = f\left(\overrightarrow{OP_3}_x - \overrightarrow{OP_1}_x\right) + \overrightarrow{OP_3}_y - \overrightarrow{OP_1}_y \\ & t_b = \frac{f\left(\overrightarrow{OP_3}_x - \overrightarrow{OP_1}_x\right) + \overrightarrow{OP_3}_y - \overrightarrow{OP_1}_y}{-f\left(\overrightarrow{OP_4}_x - \overrightarrow{OP_3}_x\right) - \overrightarrow{OP_4}_y + \overrightarrow{OP_3}_y} \end{aligned}$$

Bevor jedoch der Schnittpunkt ausgerechnet werden kann, müssen die Skalars t_a und t_b auf ihre Intervallgrenzen begutachtet werden. Liegen sie außerhalb der ihrer Intervallgrenzen liegen sie auch außerhalb der Linien. Dazu muss t_a zunächst errechnet werden:

$$t_a\left(\overrightarrow{OP_{2_x}}-\overrightarrow{OP_{1_x}}\right)-t_b\left(\overrightarrow{OP_{4_x}}-\overrightarrow{OP_{3_x}}\right)=\overrightarrow{OP_{3_x}}-\overrightarrow{OP_{1_x}} \quad | \quad +t_b\left(\overrightarrow{OP_{4_x}}-\overrightarrow{OP_{3_x}}\right)$$

$$t_a\left(\overrightarrow{OP_{2_x}}-\overrightarrow{OP_{1_x}}\right)=\overrightarrow{OP_{3_x}}-\overrightarrow{OP_{1_x}}+t_b\left(\overrightarrow{OP_{4_x}}-\overrightarrow{OP_{3_x}}\right) \quad | \quad /\left(\overrightarrow{OP_{2_x}}-\overrightarrow{OP_{1_x}}\right)$$

$$t_a=\frac{\overrightarrow{OP_{3_x}}-\overrightarrow{OP_{1_x}}+t_b\left(\overrightarrow{OP_{4_x}}-\overrightarrow{OP_{3_x}}\right)}{\overrightarrow{OP_{2_x}}-\overrightarrow{OP_{1_x}}}$$

Setzen wir nun t_b in die Geradengleichung von b ein, so erhalten wir den Ortsvektor $\overrightarrow{OS}$ was dem Schnittpunkt S entspricht.

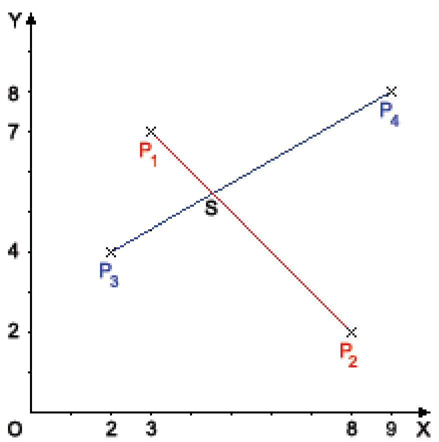
Weiterhin sollte, bevor das Skalar t_b berechnet wird, der Devisor

$$-f\,(\overrightarrow{OP_{4_x}}-\overrightarrow{OP_{3_x}})-\overrightarrow{OP_{4_y}}+\overrightarrow{OP_{3_y}}$$

Auf Null geprüft werden, denn dann wird t_b unendlich, die beiden Linien liegen aufeinander und haben somit unendlich viele Schnittpunkte.

Beispiel

Angewandt auf unser Beispiel heißt das:



Ortsvektoren:

$$\overrightarrow{OP_1}=\begin{pmatrix}3\\7\end{pmatrix};\quad \overrightarrow{OP_2}=\begin{pmatrix}8\\2\end{pmatrix};\quad \overrightarrow{OP_3}=\begin{pmatrix}2\\4\end{pmatrix};\quad \overrightarrow{OP_4}=\begin{pmatrix}9\\8\end{pmatrix}$$

Geradengleichung:

$$b: x = \overrightarrow{OP_3} + t_b\left(\overrightarrow{OP_4} - \overrightarrow{OP_3}\right)$$

$$b: x = \begin{pmatrix} 2 \\ 4 \end{pmatrix} + t_b\left(\begin{pmatrix} 9 \\ 8 \end{pmatrix} - \begin{pmatrix} 2 \\ 4 \end{pmatrix}\right) = \begin{pmatrix} 2 \\ 4 \end{pmatrix} + t_b\begin{pmatrix} 7 \\ 4 \end{pmatrix}$$

Faktor:

$$f = -\frac{\overrightarrow{OP_{2_y}} - \overrightarrow{OP_{1_y}}}{\overrightarrow{OP_{2_x}} - \overrightarrow{OP_{1_x}}}$$

$$f = -\frac{2-7}{8-3} = -\frac{-5}{5} = 1$$

Skalargleichungen:

$$t_b = \frac{f(OP_{3_x} - OP_{1_x}) + OP_{3_y} - OP_{1_y}}{-f(\overrightarrow{OP_{4_x}} - \overrightarrow{OP_{3_x}}) - \overrightarrow{OP_{4_y}} + \overrightarrow{OP_{3_y}}}$$

$$t_b = \frac{1(2-3)+4-7}{-1(9-2)-8+4} = \frac{2-3+4-7}{-9+2-8+4} = \frac{-4}{-11} = \frac{4}{11}$$

$$\frac{4}{11} \in R \cap [0;1]$$

$$t_a = \frac{OP_{3_x} - OP_{1_x} + t_b\left(OP_{4_x} - OP_{3_x}\right)}{\overrightarrow{OP_{2_x}} - \overrightarrow{OP_{1_x}}}$$

$$t_a = \frac{2-3+\frac{4}{11}(9-2)}{8-3} = \frac{-1+\frac{28}{11}}{5} = \frac{\frac{17}{11}}{5} = \frac{17}{55}$$

$$\frac{17}{55} \in R \cap [0;1]$$

Eingesetzt in Geradengleichung:

$$b: x = \begin{pmatrix} 2 \\ 4 \end{pmatrix} + t_b \begin{pmatrix} 7 \\ 4 \end{pmatrix} = \overrightarrow{OS}$$

$$\overrightarrow{OS} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} + \frac{4}{11} \begin{pmatrix} 7 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} + \begin{pmatrix} \frac{28}{11} \\ \frac{16}{11} \end{pmatrix} = \begin{pmatrix} \frac{50}{11} \\ \frac{60}{11} \end{pmatrix} \approx \begin{pmatrix} 4{,}5 \\ 5{,}5 \end{pmatrix}$$

Der Punkt S liegt laut Skizze tatsächlich in der Nähe von $(4{,}5; 5{,}5)$.

Umsetzung in BlitzMax

```
SuperStrict

Framework BRL.D3D7Max2D
Import BRL.Max2D
Import BRL.PolledInput

Type TVector2D
    Field X : Float
    Field Y : Float

    Method Add(B:TVector2D, Result:TVector2D)
        Result.X = Self.X + B.X
        Result.Y = Self.Y + B.Y
    End Method

    Method Subtract(B:TVector2D, Result:TVector2D)
        Result.X = Self.X - B.X
        Result.Y = Self.Y - B.Y
    End Method

    Method Scale(S:Float, Result:TVector2D)
        Result.X = Self.X*S
        Result.Y = Self.Y*S
    End Method

    Method Copy:TVector2D(Result:TVector2D)
        Result.X = Self.X
        Result.Y = Self.Y
    End Method

    Function Create:TVector2D(X:Float, Y:Float)
        Local Vector:TVector2D

        Vector = New TVector2D
        Vector.X = X
        Vector.Y = Y

        Return Vector
    End Function
End Type

Type TLine2D
```

```
    Field PStart : TVector2D
    Field PEnd   : TVector2D

    Method New()
        Self.PStart = New TVector2D
        Self.PEnd   = New TVector2D
    End Method

    Method Intersect:Int(B:TLine2D, Point:TVector2D)
        Local Factor:Float, Devisor:Float, TA:Float, TB:Float

        Factor = -(Self.PEnd.Y - Self.PStart.Y) / ..
                  (Self.PEnd.X - Self.PStart.X)

        Devisor = -Factor*(B.PEnd.X - B.PStart.X) - B.PEnd.Y + B.PStart.Y
        If Abs(Devisor) < 0.01 Then Return False

        TB = (Factor*(B.PStart.X - Self.PStart.X) + B.PStart.Y - ..
              Self.PStart.Y) / Devisor
        If TB < 0.0 Or TB > 1.0 Then Return False

        TA = (B.PStart.X - Self.PStart.X + TB*(B.PEnd.X - B.PStart.X)) / ..
             (Self.PEnd.X - Self.PStart.X)
        If TA < 0.0 Or TA > 1.0 Then Return False

        B.PEnd.Subtract(B.PStart, Point)
        Point.Scale(TB, Point)
        Point.Add(B.PStart, Point)

        Return True
    End Method

    Method Draw()
        DrawLine(Self.PStart.X, Self.PStart.Y, Self.PEnd.X, Self.PEnd.Y)
    End Method

    Function Create:TLine2D(PStart:TVector2D, PEnd:TVector2D)
        Local Line:TLine2D

        Line = New TLine2D
        PStart.Copy(Line.PStart)
        PEnd.Copy(Line.PEnd)

        Return Line
    End Function
End Type

Global Lines        : TLine2D[2]
Global Intersection : TVector2D

Lines[0] = TLine2D.Create(TVector2D.Create( 70.0,  50.0), ..
                          TVector2D.Create(300.0, 200.0))
Lines[1] = New TLine2D
Lines[1].PStart.X =  70
Lines[1].PStart.Y = 110

Intersection = New TVector2D

Graphics(640, 480)

While Not KeyDown(KEY_ESCAPE)
    Cls()
```

```
    Lines[1].PEnd.X = MouseX()
    Lines[1].PEnd.Y = MouseY()

    Lines[0].Draw()
    Lines[1].Draw()

    If Lines[0].Intersect(Lines[1], Intersection) Then ..
       DrawOval(Intersection.X-3, Intersection.Y-3, 7, 7)

    Flip()
Wend

EndGraphics()
End
```