

Fragen und Antworten: 3D-Vektorgraphik

Copyright © by V. Miszalok, last update: 2011-04-30

3D-Vektorgraphik

F: Zwei einfache links- und rechtshändige cartesische = rechteckige Koordinatensysteme

A: linkshändig:

li. Daumen = x-Achse waagrecht von links nach rechts

li. Zeigefinger = y-Achse senkrecht von unten nach oben

li. Mittelfinger = z-Achse = Normale auf der xy-Ebene vom Betrachter wegzeigend.

rechtshändig: wie li. aber re. Mittelfinger = z-Achse zeigt zum Betrachter.

F: Erklären Sie die 2D-Rotation und die drei 3D-Rotationen

A: 2D: Rotiert wird immer um die gedachte z-Achse mit einem Drehwinkel alpha.

3D: es gibt 3 Rotationen:

Roll = Rollwinkel = Drehwinkel um die x-Achse

(Die x-Achse von Schiffen, Autos, Flugzeugen zeigt in der Regel in Fahrtrichtung.)

Pitch = Neigungswinkel = Drehwinkel um die y-Achse

Yaw = Gierwinkel = Drehwinkel um die z-Achse

F: Vektorsumme $v_2 = v_0 + v_1$: Definition und anschauliche Deutung ?

A: Def: $v_2 = \{ v_0.X+v_1.X, v_0.Y+v_1.Y, v_0.Z+v_1.Z \}$

Deutung: v_2 zeigt vom Fußpunkt von v_0 zur Spitze von v_1 .

F: Vektordifferenz $v_2 = v_1 - v_0$: Definition und anschauliche Deutung ?

A: Def: $v_2 = \{ v_1.X-v_0.X, v_1.Y-v_0.Y, v_1.Z-v_0.Z \}$

Deutung: v_2 zeigt von der Spitze von v_0 zur Spitze von v_1 .

F: Kreuzprodukt $v_2 = v_0 \times v_1$: Definition und anschauliche Deutung ?

A: Def:

$$v_2 = v_0 \times v_1 = \begin{Bmatrix} v_0.Y \cdot v_1.Z - v_0.Z \cdot v_1.Y, \\ v_0.Z \cdot v_1.X - v_0.X \cdot v_1.Z, \\ v_0.X \cdot v_1.Y - v_0.Y \cdot v_1.X \end{Bmatrix}$$

$$v_2 = \begin{vmatrix} v_0.X & v_1.X & v_0.X & v_1.X & v_0.X & v_1.X \\ v_0.Y & v_1.Y & v_0.Y & v_1.Y & v_0.Y & v_1.Y \\ v_0.Z & v_1.Z & v_0.Z & v_1.Z & v_0.Z & v_1.Z \end{vmatrix}$$

Deutung: v_2 ist ein Vektor, der senkrecht steht auf der Ebene, die von v_0 und v_1 aufgespannt wird.

F: Skalarprodukt $v_0 \cdot v_1$: Definition und anschauliche Deutung ?

Anwendung bei Beleuchtung mit gerichtetem Licht ?

A: Def: $v_0 \cdot v_1 = v_0.X \cdot v_1.X + v_0.Y \cdot v_1.Y + v_0.Z \cdot v_1.Z = |v_0| \cdot |v_1| \cdot \cos(\alpha)$

Deutung: Wenn v_0 und v_1 beide die Länge 1.0 haben,

dann ist $v_0 \cdot v_1$ der Cosinus des Winkels zwischen beiden.

Unter gerichtetem Licht bestimmt das momentane Skalarprodukt =

$|Normale| \cdot |Lichtvektor| \cdot \cos(\alpha)$

(wobei die Beträge $|Normale|$ und $|Lichtvektor|$ in der Regel = 1.0 sind) die Helligkeit der Fläche.

F: Was versteht man unter der Normalen einer 3D-Fläche ? Verwendung ? Zweck der Vertexnormalen ?

A: Lotvektor auf der Fläche. Richtung zur Außenwelt, Betrag meistens normiert auf 1.0,

Fußpunkt ohne Bedeutung (wird weggelassen).

Verwendung: a) Unterscheidung von Vor- und Rückseite b) Berechnung der Reflektion von gerichtetem Licht

In DirectX3D und in OpenGL gibt man nicht jeder Fläche (Face),

sondern jedem Vertex eine Normale mit folgendem Zweck: Mittelung der zusammenstoßenden Eckennormalen verhindert Helligkeitssprünge an Ecken und Kanten (Gouraud-Shading).

F: Eine Fläche steht unter gerichtetem Licht. Der Winkel zwischen der Normale und dem Lichtvektor sei α . Bestimmen Sie $\cos(\alpha)$ und die Helligkeit der Fläche.

alpha [Grad]	0	60	90	120	180	360
cos(alpha)						
Helligkeit						

A:

alpha [Grad]	0	60	90	120	180	360
cos(alpha)	1.0	0.5	0.0	-0.5	-1.0	1.0
Helligkeit	100%	50%	0%	0%	0%	100%

F: Was ist ein TriangleStrip ? Vorteil, Nachteile ? Geschlossener und offener TriangleStrip ?

A: Ein `Vector3`-Array, das ein kontinuierliches Band von Dreiecken definiert, wobei, ausgehend von einem Anfangsdreieck p_0, p_1, p_2 jede folgende Ecke p_i ein neues Dreieck mit der Vorgängerkante aufspannt.

Vorteil: sehr kompakte Flächendefinition, ideal für Wand von z.B. Zylinder, Pyramide.

Nachteile: 1) Reihenfolge der Ecken ist schwierig.

2) ungeeignet für allseitig geschlossenen Körper wie z.B. Würfel, Kugel.

Der TriangleStrip ist geschlossen, wenn die letzte Ecke identisch mit der allerersten und die zweitletzte identisch mit der zweiten ist, ansonsten ist er offen.

F: Was ist eine TriangleList ? Vorteile, Nachteil ?

A: Ein `Vector3`-Array, das Vertices in beliebiger Reihenfolge codiert = `VertexArray` plus ein Array aus Nummertriplets = `IndexArray`, wo jedes Triplet je drei Vertexnummern i, j, k aus dem `VertexArray` enthält.

Vorteile: 1. Jeder Vertex wird nur einmal codiert. 2. Reihenfolge der Dreiecke ist frei wählbar.

Nachteil: Die zweistufige Datenstruktur wird leicht unübersichtlich.

F: Was versteht man unter Vertexformat ? Beispiele

A: Vertexformat = Datenstruktur zum Speichern von Eigenschaften eines 3D-Eckpunkts

`CustomVertex.PositionOnly` = nur X,Y,Z ohne Erweiterung

`CustomVertex.PositionColored` = X,Y,Z plus RGB-Farbwert

`CustomVertex.PositionNormal` = X,Y,Z plus Normale N_x, N_y, N_z

`CustomVertex.PositionNormalTextured` = X,Y,Z, N_x, N_y, N_z plus Texturpunkt T_u, T_v

F: Was versteht man unter 1) VertexArray mit Beispiel 2) VertexBuffer ?

A:

1) Array von Vertices im Hauptspeicher: Beispiel:

```
CustomVertex.PositionColored[] vv = new CustomVertex.PositionColored[3];
```

2) `Direct3D`-Klasse, die ein `VertexArray` via Bus in den Input-RAM der Graphikkarte transportiert.

F: Was versteht man unter 1) IndexArray mit Beispiel 2) IndexBuffer ? Zwecke ?

A: 1) Hauptspeicher-Array zusammengesetzt aus Triplets mit je drei Vertexnummern:

```
Int16[] ii = { 0,1,2, 1,2,3 };
```

2) `Direct3D`-Klasse, die ein `IndexArray` via Bus in den Input-RAM der Graphikkarte transportiert

Zwecke: a) Befreit den `VertexBuffer` vom Zwang zur starren Reihenfolge der Vertices.

b) Jeder Vertex kann mehrfach in vielen Dreiecken verwendet werden.

F: Zeichnen Sie folgende drei Primitive Meshes:

1) ein 5-Eck mit dem Umfang 5f,

2) einen Würfel mit der Kantenlänge 1f und

3) eine Kugel mit dem Radius 1f und 20 Längen- und 20 Breitengraden

A:

```
Mesh fuenfeck = Mesh.Polygon( device, 1f, 5 );
Mesh wuerfel  = Mesh.Box    ( device, 1f, 1f, 1f );
Mesh kugel    = Mesh.Sphere ( device, 1f, 20, 20 );
device.BeginScene();
    fuenfeck.DrawSubset( 0 );
    wuerfel  .DrawSubset( 0 );
    kugel    .DrawSubset( 0 );
device.EndScene();
```

F: Was ist ein X-File ? Mindestinhalt ?

A: populäres Format für Speicherung/Transport von 3D-Daten = DirectX-File-Format = DXF.

Zwei Formate: 1) binär, kompakt, unlesbar 2) editierbarer Text

Headerzeile beginnt mit xof.

Es folgen `Templates` (nicht obligatorisch) und `Data` (obligatorisch).

`Data` enthält mindestens ein `Mesh` mit `VertexArray` und `IndexArray` und eine `MeshMaterialList`.