

OpenGL vs. Direct3D

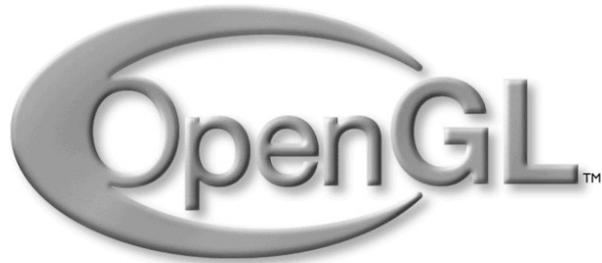
Ein 3D-API Vergleich

3D-APIs (Application Programming Interface) sind Software-Schnittstellen, um über einen Standard auf die Hardware des Rechners zugreifen zu können. Direct3D von Microsoft und der offene Standard OpenGL sind die wohl meistgenutzten 3D-APIs.

OpenGL kommt aus der professionellen Ecke des CAD und der wissenschaftlichen Visualisierung, während Direct3D meist bei Spielen eingesetzt wird. Während OpenGL auf fast allen gängigen Plattformen verfügbar ist, glänzt Direct3D mit DirectX, einer ganzen Anzahl anderer APIs, z. B. für Sound.

Übersicht:

- Was leistet eine 3D-API
- Die 3D-Hardware
- OpenGL
- Direct3D
- Fahrenheit
- OpenGL, Direct3D – Vor- und Nachteile
- Synthese und Empfehlung



Was leistet eine 3D-API?

Was ist eine 3D-API?

Eine 3D-API bildet eine Schnittstelle zur 3D-Hardware.

Da man sich meist nicht auf eine spezielle Grafikkarte einschränken will, benutzt man eine spezielle API, um auf die Hardware zuzugreifen.

APIs erleichtern meist auch die Programmierung, indem sie abstraktere Modelle zur Verfügung stellen, die für den Programmierer intuitiver sind.

OpenGL und Direct3D sind hardwareunabhängig, so dass sie mit verschiedener oder auch ohne 3D-Hardware zur Verfügung stehen.

Sowohl Direct3D als auch OpenGL können Funktionen die sie implementieren, die aber von der Hardware nicht unterstützt werden emulieren.

Welche 3D-APIs gibt es?

- OpenGL
- Direct3D
- Quickdraw3D (Apple)
- Glide (3DFx)
- Java3D (Sun, benutzt OpenGL)
- Mesa (OpenGL ähnlich)
- Warp3D (Amiga)

Die 3D-Hardware

3D-Hardware kann im Wesentlichen Dreiecke zeichnen.

Den Eckpunkten dieser Dreiecke können Farbwerte zugewiesen werden.

Wenn alle drei Eckpunkte den gleichen farbwert haben, so erscheint das Dreieck in dieser Farbe.

Weist man den Eckpunkte unterschiedliche Farben zu, rendert die Hardware ein Dreieck mit einem Farbverlauf.

Des weiteren ist es möglich dem Dreieck eine Textur zuzuweisen, indem man Jedem Vertex eine Koordinate auf einer Textur zuweist.

Diese Textur ist ein Bitmap, welches dann auf das Dreieck gelegt wird.

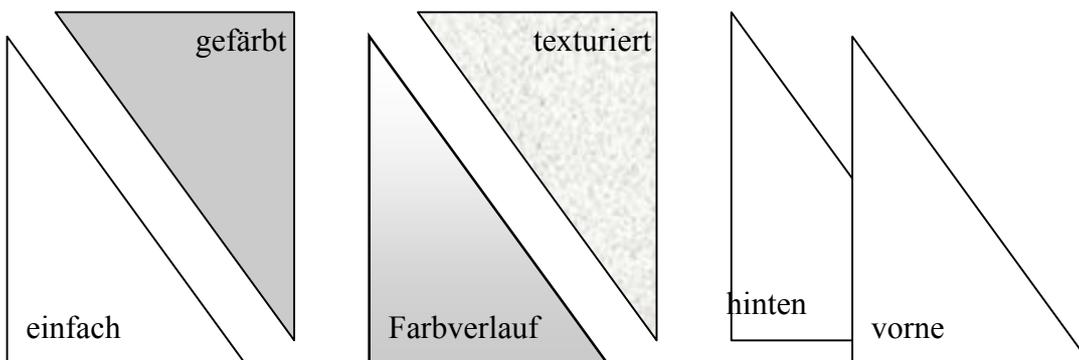
Es ist auch eine Kombination aus Farbe und ev. mehreren Texturen möglich.

Außerdem implementiert die Grafikhardware eine Tiefeninformation der gegebenen Dreiecke (eigentlich der Eckpunkte).

Dadurch kann die Hardware beim Rendern feststellen, welche Punkte sich vor anderen befinden, und so nur die Punkte darstellen, die nicht verdeckt werden.

Zuletzt sind noch die Per-Fragment-Operations möglich, die schon gerenderte Pixel verändern, z. B. um halbdurchsichtige Dreiecke zu zeichnen.

Die 3D-Hardware, Beispieldreiecke



OpenGL

Geschichte:

- 80er Jahre IrixGL von SGI (nur für Irix)
- 1992 OpenGL 1.0
- 1995 OpenGL 1.1
- 1998 OpenGL 1.2

OpenGL ist eine Weiterentwicklung von IrixGL, das SGI für Irix, sein eigenes Unix-Betriebssystem, entworfen hat.

Kontrolle über OpenGL

Als offener Standard, gehört OpenGL nun nicht mehr SGI alleine, sondern wird von der ARB (Architecture Review Board) verwaltet. Mitglieder in der ARB sind nVIDIA, SGI, DEC, IBM, Intel, Microsoft und andere.

Das ARB trifft sich viermal jährlich, um über den Standard zu beraten und zu beschließen, wie es weitergehen soll.

Die Protokolle der Sitzungen sind unter www.opengl.org öffentlich zugänglich.

Unterstützte Betriebssysteme:

OpenGL ist auf folgende Plattformen implementiert:

AIX, Irix, FreeBSD, NetBSD, OpenBSD, Linux, Windows 9x, Windows NT, OS/2, Amiga OS, BeOS, MacOS und andere Unix-Betriebssysteme.

Damit kann man sagen, dass OpenGL betriebsystemunabhängig ist.

Jedes Betriebssystem ist selbst für die Implementierung verantwortlich.

Die ARB zertifiziert diese lediglich. Allerdings existiert eine Open-Source Referenzimplementation.

Weiterhin implementiert OpenGL das Client-Server Modell von X, das es erlaubt, die graphische Ausgabe von der Datenberechnung zu trennen.

Damit kann man z. B. über ein Netzwerk OpenGL-Programme auf einem entfernten Server laufen lassen, und die Ausgabe auf den lokalen Computer umleiten.

Beispielprogramm:

```
glBegin (GL_TRIANGLES);  
    glVertex (0,0,0);  
    glVertex (1,1,0);  
    glVertex (2,0,0);  
glEnd ();
```

Dieses Programm erzeugt ein planes Dreieck.

Die Eckpunkte sind durch die Koordinaten hinter ‚glVertex‘ gegeben.

OpenGL Utility Toolkit – GLUT:

Das GLUT ist eine Erweiterung für die Programmierung mit OpenGL In C oder Fortran.

Sie stellt abstraktere Programmiermöglichkeiten zur Verfügung. Außerdem ermöglicht sie eine plattformunabhängige Ansteuerung des jeweiligen Window-Systems.

So kann man per GLUT ein Fenster für das OpenGL-Programm im Source-Code anfordern, und die GLUT sorgt bei der Kompilierung an die Angleichung dieses Codes an das jeweilige Window-Systems.

Beispielanwendungen:

- ArchiCAD
- Autodesk (AutoCAD 2000)
- Quake (GL-Quake, Quake II & III)
- Microstation 95
- MathGL3D (Mathematica)
- Maya (Character animation, Modeling)

Direct3D

Geschichte von Direct3D

- März 1996 DirectX 2.0
- September 1996 DirectX 3.0
- September 1997 DirectX 5.0
- Juli 1998 DirectX 6.0
- Mai 1999 DirectX 7.0
- November 2000 DirectX 8.0

Kontrolle über Direct3D:

Direct3D ist als Bestandteil von DirectX alleiniges Eigentum von Microsoft.

Ev. In Zusammenarbeit mit den Hardwareherstellern, beschließt Microsoft die Änderungen in der API allein.

Direct3D und DirectX

DirectX enthält außer Direct3D noch weitere Schnittstellen um andere Hardware anzusprechen.

z. B. DirectSound, DirectPlay, DirectInput;
die macht DirectX und damit Direct3D für Spielentwickler interessant, da diese meist auch Sound, Musik, Joysticks, etc. benutzen.

Unterstützte Betriebssysteme:

Direct3D unterstützt lediglich die Windows-Plattform.

DirectX 8.0, und damit auch Direct3D 8.0 gibt es allerdings nur für Windows 9x und Windows 2000.

Für DOS-basierte Windows Versionen und für Windows NT Bis Version 3.51 gibt es kein DirectX.

Windows NT 4.0 unterstützt höchstens Direct3D 3.0, allerdings

nur ohne hardware-beschleunigung.

Beispielprogramm Direct3D

```
D3DVERTEX g_pvTriangleVertices[3];

D3DVECTOR p1( 0.0f, 3.0f, 0.0f );
D3DVECTOR p2( 3.0f,-3.0f, 0.0f );
D3DVECTOR p3(-3.0f,-3.0f, 0.0f );

g_pvTriangleVertices[0] = D3DVERTEX( p1, vNormal, 0, 0 );
g_pvTriangleVertices[1] = D3DVERTEX( p2, vNormal, 0, 0 );
g_pvTriangleVertices[2] = D3DVERTEX( p3, vNormal, 0, 0 );

pd3dDevice->DrawPrimitive( D3DPT_TRIANGLELIST,
                          D3DFVF_VERTEX, g_pvTriangleVertices, 3, NULL
```

Dieses Programm erzeugt ein planes Dreieck.
Die Eckpunkte sind durch die Koordinaten hinter ‚D3DVECTOR‘ gegeben.

Immediate & Retained Mode:

Direct3D besitzt zwei Modi:
Immediate & Retained Mode.
Immediate ist hardwarenah, Retained abstrakter und mehr Model-orientiert.
Retained wurde mit Direct3D 7.0 aufgegeben.

Beispielanwendungen:

- Simply 3D
- Unreal Tournament
- Earth 2150
- Warcraft III
- Microsoft Flight Simulator
- Battle Isle 4
- ...

Fahrenheit:

Fahrenheit war eine Initiative von Microsoft und SGI:
Es sollte eine neue 3D API entstehen, die OpenGL und Direct3D miteinander verschmelzen soll. (zeitlich ca. mit DirectX 9.0)
Das Projekt ist mittlerweile gescheitert.
Sowohl SGI als auch Microsoft hatten kein Interesse mehr einen gemeinsamen Standard zu etablieren.

OpenGL, Vor- und Nachteile

Vorteile von OpenGL

- plattformunabhängig
- selbst erweiterbar
- Open-source Referenzimplementierung
- Client-Server-Model
- Meist bessere Treiber für professionelle Grafikkarte

Nachteile von OpenGL

- Neuer Standard dauert lange
- Extensions-Chaos
- Keine bzw. nur schlechte Treiberunterstützung auf preiswerter Hardware

Direct3D, Vor- und Nachteile

Vorteile von Direct3D

- Schnell neuer Standard verfügbar
- Programmiersprachenunabhängig (COM)
- DirectX immer auch verfügbar (Sound...)
- Standard meist weiter als Hardware-entwicklung
- Software-Emulator für alle Features
- Meist bessere Treiber für Low-Cost Grafikkarten

Nachteile von Direct3D

- Plattformabhängig (Windows)
- Proprietär (Microsoft)
- Closed-source
- Oft starke Änderungen bei Versions-wechseln

Synthese und Empfehlung

OpenGL wird in der Wissenschaft und bei professionellen Anwendungen bevorzugt.

Viele CAD-Arbeitsplätze sind z. B. an Unix-Workstations, sodass dort sowieso keine Wahl besteht. Auch viele Wissenschaftler arbeiten nicht mit Windows NT.

Für ausgewachsene professionelle Grafikkarten gibt es aber meist auch keine Direct3D-Treiber.

Direct3D wird meist bei Spielen eingesetzt, da dort sowieso zur Zeit nur für Windows entwickelt wird und fast immer auch Sound und ähnliches benötigt wird, was durch DirectX realisiert wird.

Beide haben ihre Vor- und Nachteile, es ist keine allgemeingültige eindeutige Empfehlung möglich