

Frameworks: Agenda

- Aufgabe von Medien-Frameworks

- Microsoft DirectX
- SDL

- Microsoft Windows Media
- Java Media Framework

- Vorführung: DirectX SDK Samples
- Vorführung: WindowsMedia9 HD

DirectX: Literatur

www.microsoft.com/directx

(DirectX portal page)

www.microsoft.com/msdn

(MS developer network)

DirectX SDK

(Docs und Beispiele)

Bargen, Donnelly: Inside DirectX, Microsoft Press, 1998

Root, Boer, DirectX complete, McGraw-Hill, 1999

Sirotnin, Debeloff, Urri: DirectX-Programmierung mit Visual C++, Addison-Wesley, 1999

Petzold, Programming Windows95, Microsoft Press, 1996

Stroustrup, C++ Programming Language, Addison-Wesley, 1997

Fosner, Real-Time Shader Programming with DirectX9, Morgan-Kaufmann, 2003

Scherfgen, 3D-Spieleprogrammierung (DirectX9), Hanser, 2003

c't Artikelserie zu Delphi mit DirectX, 1999 bis 2000

DirectX: Übersicht

Direct<X>: Multimedia-Schnittstellen für Windows

- DirectDraw high-performance 2D-Graphik
- Direct3D 3D-Graphik
- DirectShow Videowiedergabe

- DirectSound grundlegende Audiofunktionen (3D)
- DirectMusic Software-Synthesizer

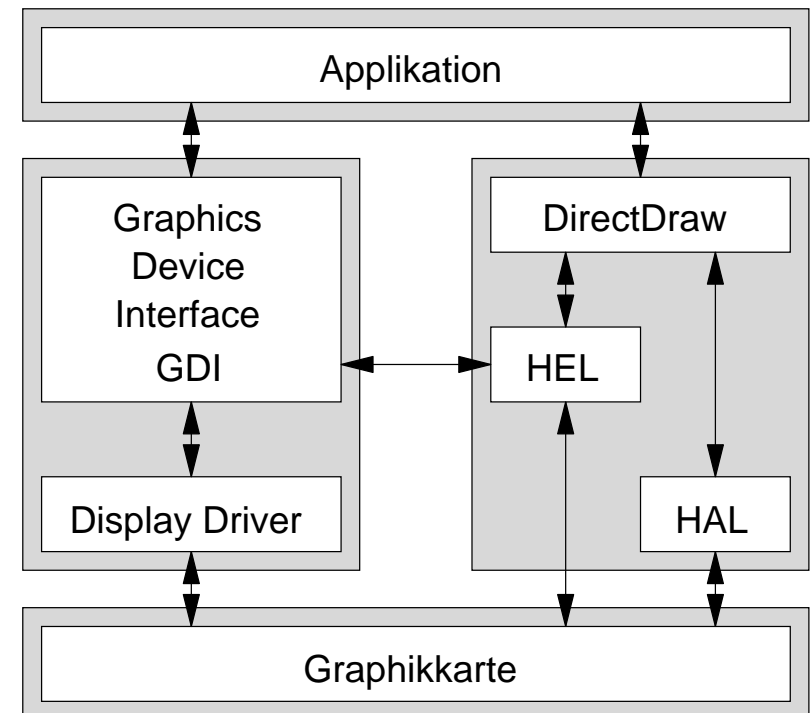
- DirectInput schnelle Treiber für Tastatur/Maus/Joystick/...
- DirectSetup einfache und sichere Installation
- DirectPlay Multiplayer / Internet
- Speech SDK Spracheingabe und -ausgabe

- was fehlt? . . .

DirectX: DirectDraw

DirectDraw: Grundfunktionen für 2D-Graphik

- direkte (low-level) Verwaltung des Bildspeichers
- front buffer / back buffer / surface flipping
- hardware overlays
- bit-blitting
- kein Ersatz für das GDI:
Linien, Fonts, etc. per GDI
- HEL: Hardware Emulation Layer
- HAL: Hardware Abstaction Layer
- DDI: GDI Device Driver Interface
- "FoxBear" Demo



DirectX: Direct3D

*"world-class game and interactive 3D-graphics
on a computer running Microsoft windows"*

(MS DirectX overview)

- geräteunabhängiger Zugriff auf 3D-Hardware
- nutzt Hardwarebeschleunigung, soweit vorhanden
- oder Softwareemulation

- Transformation, Clipping
- Z-Buffer / W-Buffer
- Rendering mit Flat- / Gouraud-Shading
- Texturen, Mip-Mapping
- diverse Lichtquellen

- Funktionsumfang vergleichbar mit OpenGL (seit DirectX7)
- programmierbare "Shader" und HLSL-Sprache (seit DX8)

DirectX: DirectShow

"streaming media architecture"

- Wiedergabe und Aufnahme von Datenströmen
- Formatkonvertierung
- komprimierte Audio/Videodaten
- verwendet Hardwarefunktionen, soweit vorhanden

- WAV, MP3, ...
- MPEG, AVI, ...

- basiert auf WDM-Treibern
- aber auch Unterstützung von "legacy"-Treibern (video for windows)

- in DirectX9 überarbeitet, Wiedergabe jetzt über 3D-Pipeline

DirectX: DirectSound

Audiowiedergabe:

- flexibler Audiomixer
- nutzt Audiohardware soweit möglich, sonst Software
- schnell, geringe Latenz
- Sampleraten-/Formatkonvertierung

- 3D-Audiofunktionen
- Position und Richtung von Hörer und Schallquellen
- Doppler-Effekt, entfernungsabhängige Dämpfung
- HRTF-Funktionen

- seit DirectX9 bis 192 KHz, mehrkanalfähig
- weitere Details: Vorlesung PC-Technologie

DirectX: DirectMusic

"message-based musical data"

- Noten, Akkorde, Phrasierung, ...
- MIDI-Schnittstelle
- Klangerzeugung über Software-Synthesizer
- und Wiedergabe über DirectSound
- oder Ansteuerung externer Synthesizer

- DLS (downloadable sounds / SoundFont2)
- DirectMusic Producer

- Hardwareunterstützung ab Windows98/2000

DirectX: DirectInput

- Treiberschicht zu Eingabe-Geräten
- Treiber für sehr viele Geräte
- umgeht die normalen Windows-Treiber
- minimale Latenz

- Unterstützung von force-feedback Geräten
- diverse Krafteffekte / -kennlinien

- vgl. Immersion iForce Studio

3D-Graphik: Prinzip

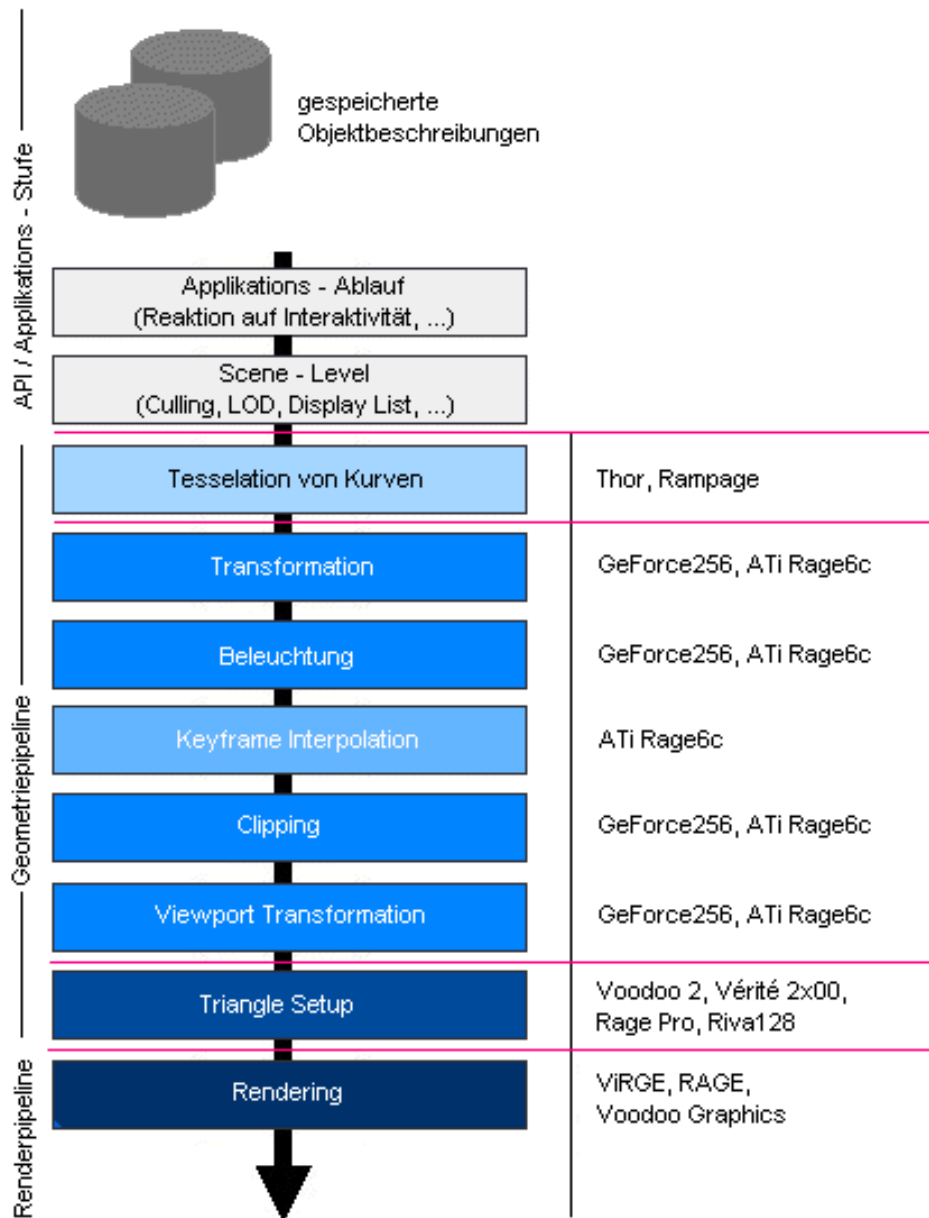
Standard-Prinzip für 3D-Graphik:

- 3D-Modellierung der Objekt-Oberflächen (statt Volumen)
- über Polygone oder Freiformflächen (z.B. Beziersplines)
- Umsetzung in Dreiecksnetze
- Perspektiv-Koordinatentransformation

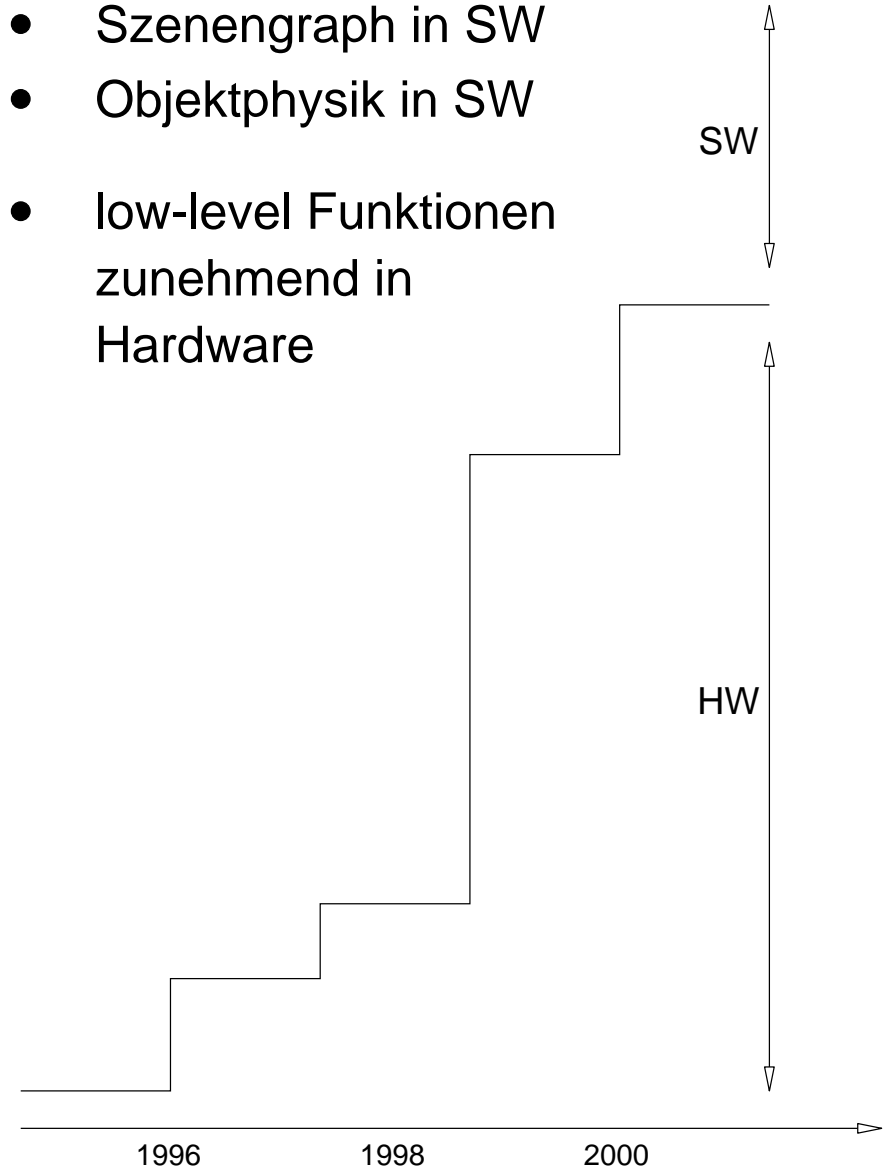
- Beleuchtungsberechnung für jedes einzelne Dreieck
- "aufgeklebte" Texturen für jedes einzelne Dreieck
- Zerlegen der Dreiecke in Rasterlinien
- Berechnung der Bildfarben für jeden Punkt auf jeder Linie
- Sichtbarkeitstest über "z-buffer"

- extrem rechenaufwendig, aber gut parallelisierbar
- Spezialhardware: GPU "graphics processing units"

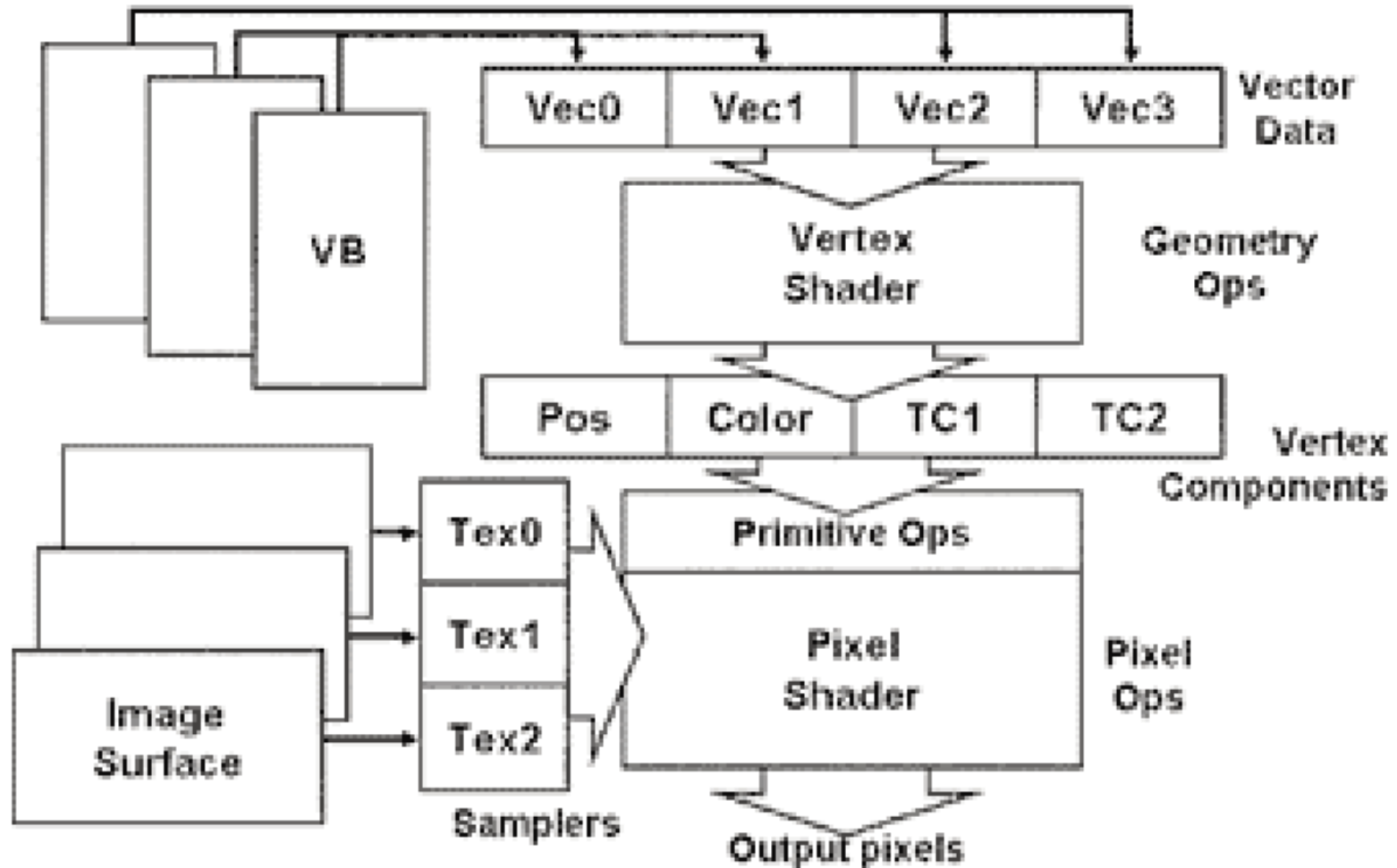
Graphik: 3D-Pipeline



- Szenengraph in SW
- Objektphysik in SW
- low-level Funktionen zunehmend in Hardware

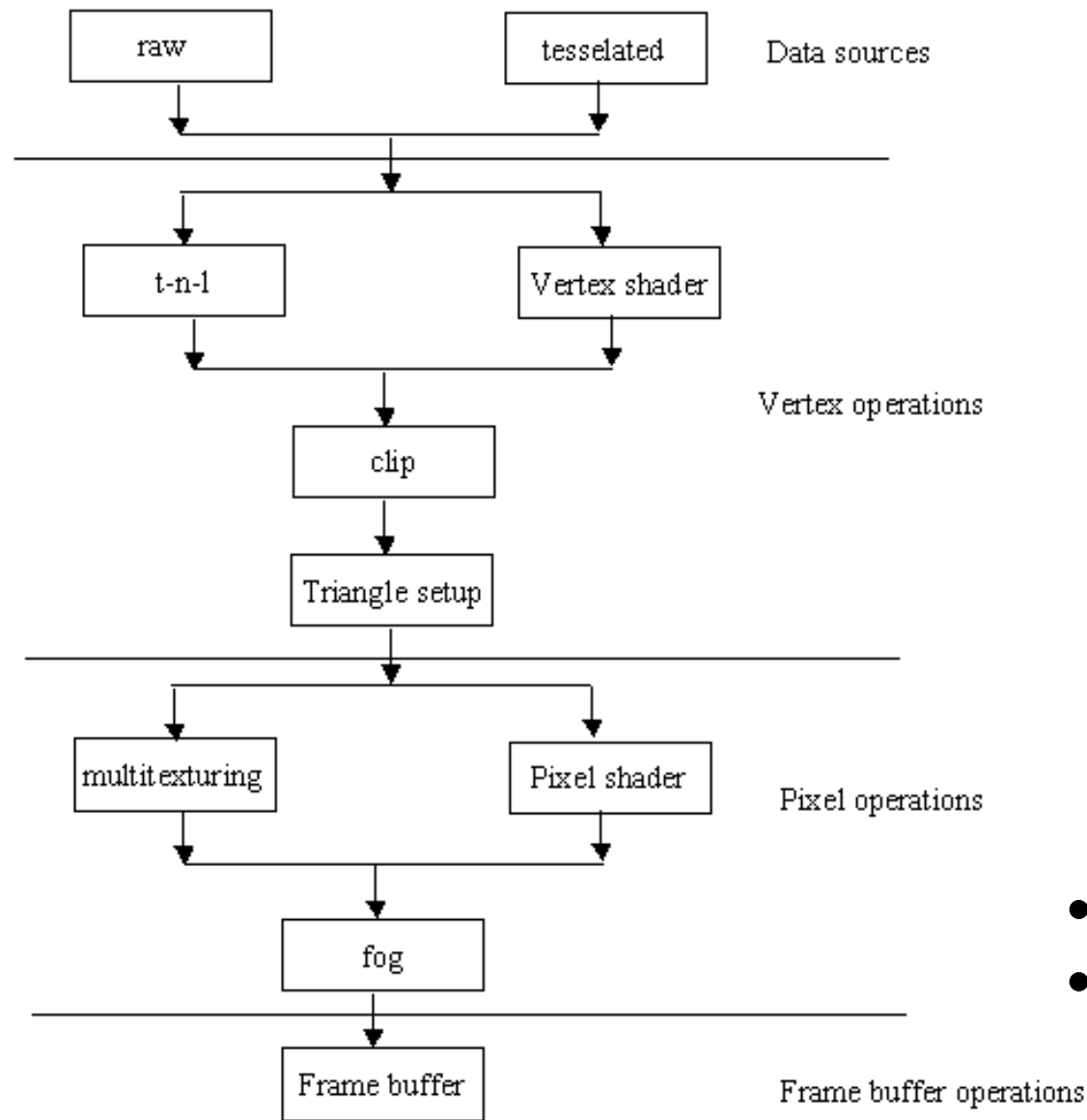


DirectX8: "shader"



- kombiniert bisherige APIs: DirectDraw und Direct3D
- 3D-Pipeline teilweise programmierbar

DirectX 8: Graphik-Pipeline



- kompatibel mit DirectX 7
- neue Fkt. nur auf "neuer" HW

DirectX 8: Beispiele für VertexShader



- morphing / twining animation
- matrix palette skinning
- user-defined lighting models

- general environment mapping
- procedural geometry
- developer defined algorithms

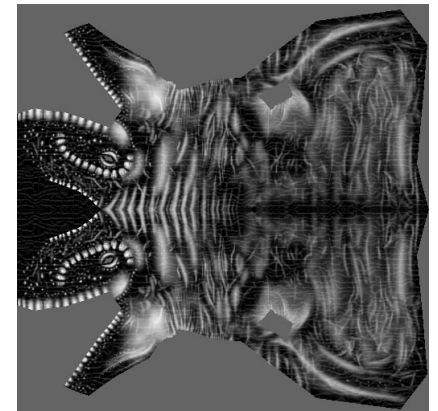
DirectX 8: *PixelShader*

PixelShader: Programmiersprache für Pixel-Ops:

- direkter Zugriff auf die Graphik-Hardware,
- insb. für Textur-Operationen:

- beliebige Textur-Operationen
- per-pixel lighting
- bump-mapping
- per-pixel environment mapping
- other developer-defined algorithms

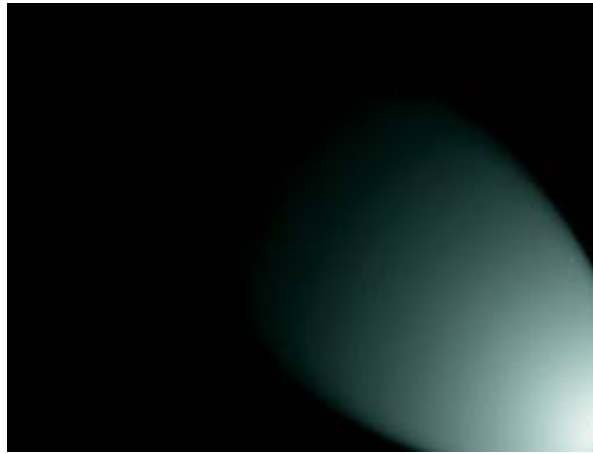
- in DirectX9 nochmal deutlich erweitert
- in Zusammenarbeit mit ATI und nVidia



Direct3D: Shader Beispiele

- externe Folien, MS DX8 Overview (PowerPoint)
- ATI "wood shader"

DirectX 8: Multitexturing



- Überlagerung mehrerer Texturen
- z.B. für light-maps (siehe oben)
- oder bump-mapping (rechts)

Ati Radeon: 2 Pipes, je 3 Texturen

Geforce 2: 4 Pipes, je 2 Texturen



DirectX 8/9: *HLSL*

"High Level Shader Language"

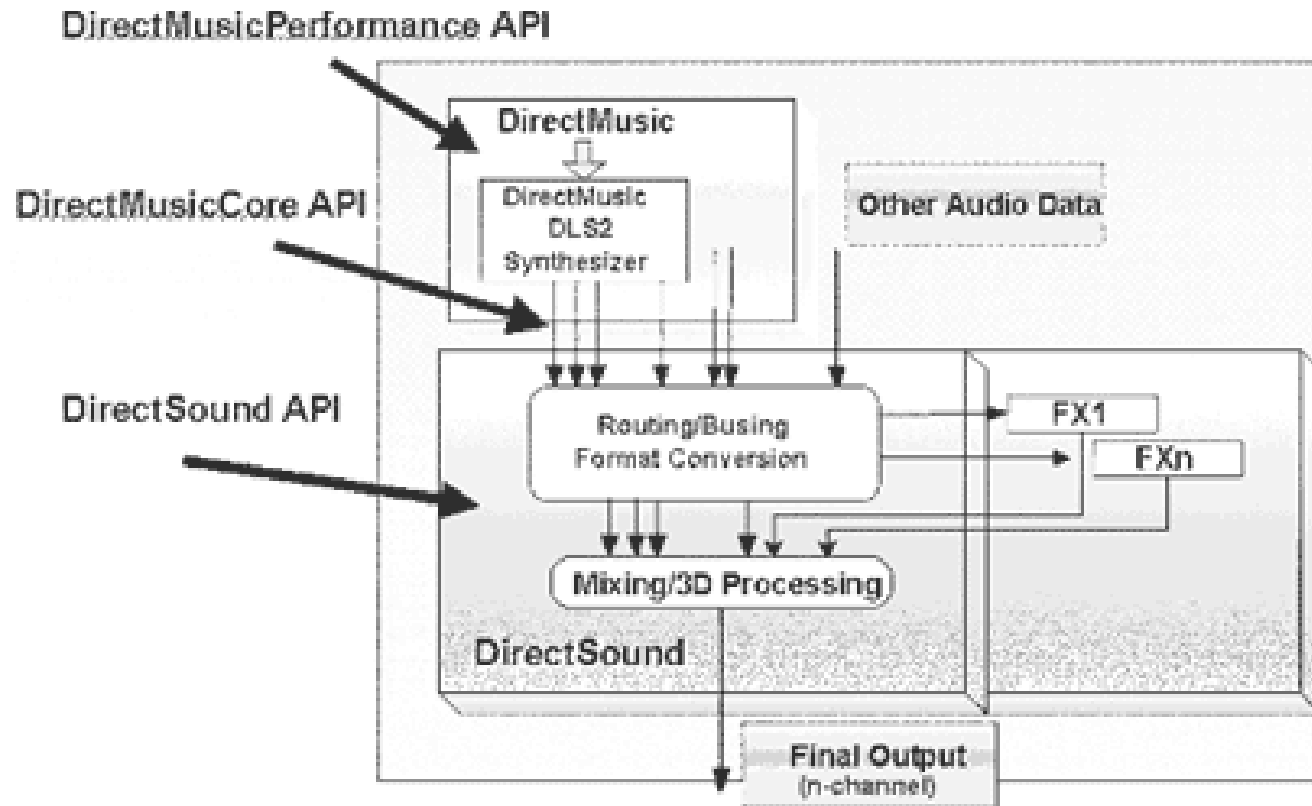
- C-ähnliche Programmiersprache für Shader
- Compiler erzeugt effizienten Code für Graphikchips

diverse Technologiebeispiele von ATI und nVidia:

- realistisch aussehende Haut
- 3D-Holzmaserung
- spiegelnde und brechende Glasflächen
- begegte, spiegelnde und brechende Wasseroberflächen
- Partikeleffekte, Feuer, ...

- siehe: "GPUGems"-Buch, GDC-2004 Präsentationen

DirectX 8: Audio

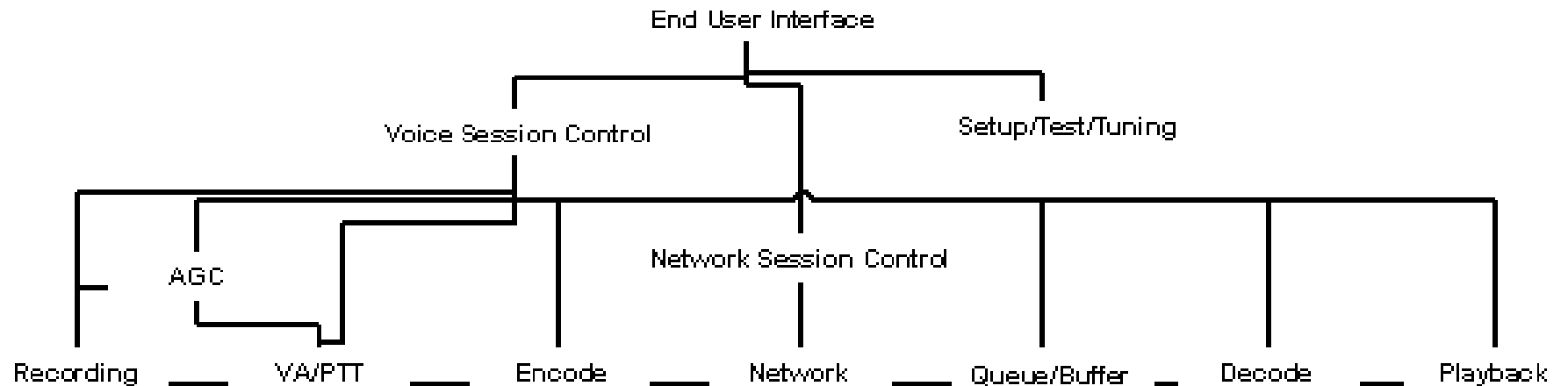


Integration von DirectSound und DirectMusic:

- 3D-Audio für alle Quellen (.wav, .asf, DLS, ...)
- DLS2 SW-Synthesizer (6-stage envelopes, effects, LFOs, ...)

DirectX 8: Voice Communication

DirectPlayVoice: Echtzeit-Sprachkommunikation in DirectPlay



- Verwaltung über den DirectPlay Server
- automatische Verwaltung jeder DirectPlay Voice Session
- automatische Kalibrierung der Bandbreite/Latenz

DirectX 8/9: DirectShow & DVD

- DirectX 8 Videobeschleunigung
- volle Unterstützung von Windows Media (ASF/WMV)
- im Prinzip MPEG-2 Unterstützung inkl. DVD-Player

- in DirectX 9 wieder komplett überarbeitet:
- "Video Mixing Renderer Filter VMR-9"
- erstmals Unterstützung für De-Interlacing
- erstmals Unterstützung für DVB-T/-S/-C
- erfordert Windows XP

- WMV-HD Demos

DirectX: COM

Component Object Model:

- Microsoft's Komponentenmodell
- entwickelt für OLE (object linking and embedding)

- Interaktion zwischen Software-"Objekten" (Komponenten)
- über Schnittstellen (Interfaces)
- als Tabelle mit Zeigern auf C-Funktionen

- Interface-Hierarchie, IUnknown
- Schnittstelle wird nach Definition nie mehr verändert (GUID)
- statt dessen Einführung neuer Schnittstellen
- Speicherverwaltung mit Referenzzählung

`QueryInterface()` / `AddRef()` / `Release()`

DirectX: vtable

COM-Funktionsaufrufe:

- vtable: Tabelle mit Funktionspointern
- COM-Referenz ist Pointer auf eine vtable
"long pointer to Direct Draw interface"

```
lpDD -> lpVtbl ->Release( lpDD );           /* C */  
lpDD -> Release();                          // C++
```

- keine Ausnahmebehandlung
- Fehlercodes jeder Funktion prüfen!

```
if (FAILED( lpDDSBack->GetDC( &hdc )) { ... }  
if (SUCCEEDED( lpDDSBack->GetDC( &hdc )) { ... }
```

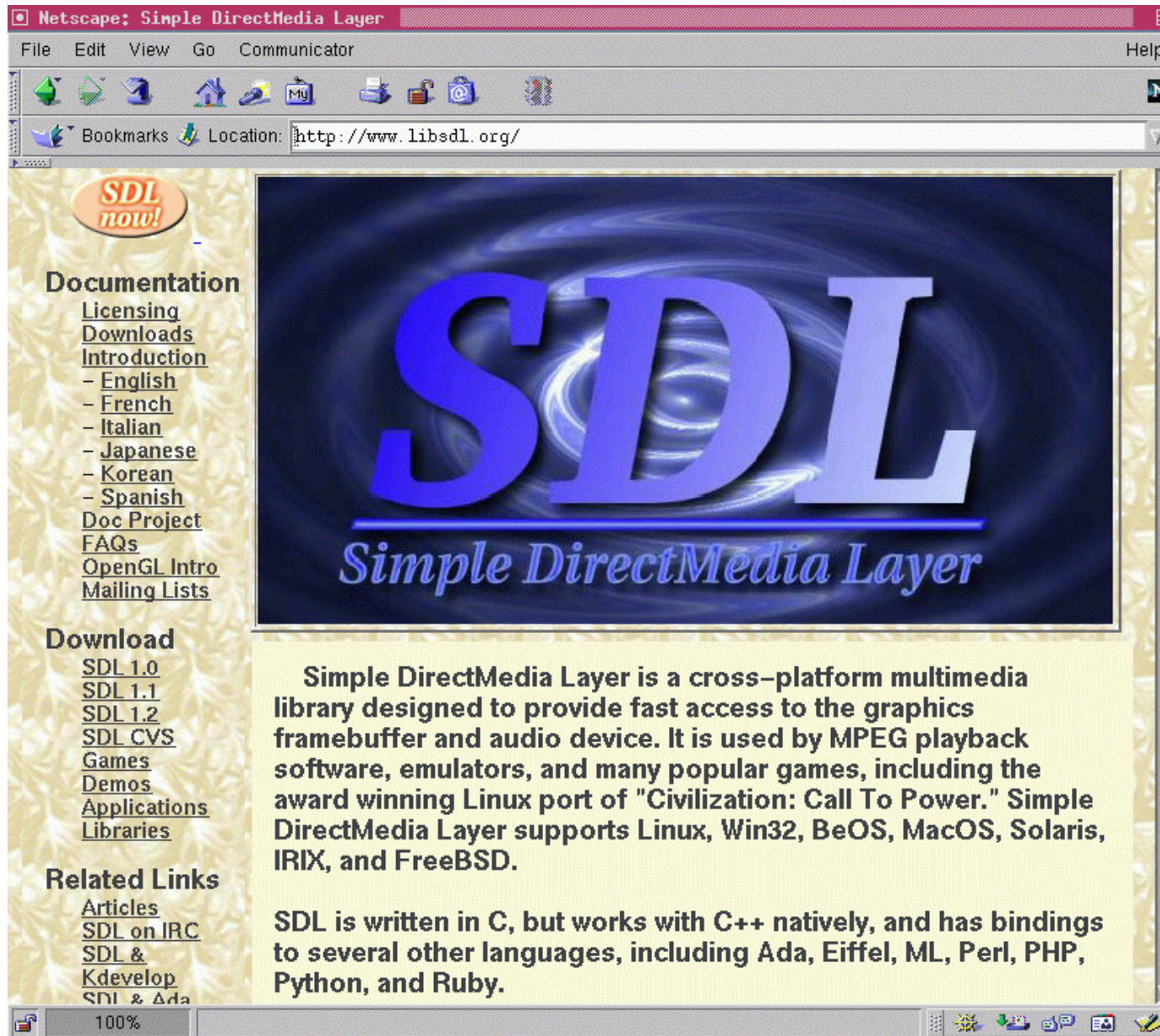
DirectX 9: Managed DirectX

DirectX 9 erlaubt Integration in .NET-Framework:

- Schnittstelle von DirectX-Funktionen nach .NET
- Nutzen der .NET Klassenbibliothek aus DirectX-Applikationen
- Speicherverwaltung über .NET
- Ausnahmebehandlung über .NET
- für alle .NET-Sprachen: neue Beispiele in C#, VB.NET, ...

- Performanceanforderung: mindestens 90% der Leistung von nativen C++

SDL: "Simple DirectMedia Layer"



The screenshot shows a Netscape browser window with the title "Netscape: Simple DirectMedia Layer". The address bar contains "http://www.libsdl.org/". The main content area features a large blue "SDL" logo with the text "Simple DirectMedia Layer" underneath. To the left of the logo is a navigation menu with sections: "Documentation" (Licensing, Downloads, Introduction, English, French, Italian, Japanese, Korean, Spanish, Doc Project, FAQs, OpenGL Intro, Mailing Lists), "Download" (SDL 1.0, SDL 1.1, SDL 1.2, SDL CVS, Games, Demos, Applications, Libraries), and "Related Links" (Articles, SDL on IRC, SDL & Kdevelop, SDL & Ada). Below the logo, there is a paragraph of text: "Simple DirectMedia Layer is a cross-platform multimedia library designed to provide fast access to the graphics framebuffer and audio device. It is used by MPEG playback software, emulators, and many popular games, including the award winning Linux port of 'Civilization: Call To Power.' Simple DirectMedia Layer supports Linux, Win32, BeOS, MacOS, Solaris, IRIX, and FreeBSD." Below this is another paragraph: "SDL is written in C, but works with C++ natively, and has bindings to several other languages, including Ada, Eiffel, ML, Perl, PHP, Python, and Ruby." The browser status bar at the bottom shows "100%" zoom.

(www.libsdl.org)

SDL: Beispiel

```
/* Print out all the keysyms we have, just to verify them */

#include <stdio.h>
#include <ctype.h>

#include "SDL.h"

int main(int argc, char *argv[])
{
    SDLKey key;

    if ( SDL_Init(SDL_INIT_VIDEO) < 0 ) {
        fprintf(stderr, "Couldn't initialize SDL: %s\n",
            SDL_GetError());
        exit(1);
    }
    for ( key=SDLK_FIRST; key<SDLK_LAST; ++key ) {
        printf("Key #%d, \"%s\"\n", key, SDL_GetKeyName(key));
    }
    SDL_Quit();
    return(0);
}
```