

DirectX: Agenda

- Einführung und Konzepte
- Vorführung DirectX7 SDK

PC-Technologie | SS 2001 | 18.214

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

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

PC-Technologie | SS 2001 | 18.214

DirectX: Übersicht

Direct<X>: Multimedia-Schnittstellen für Windows

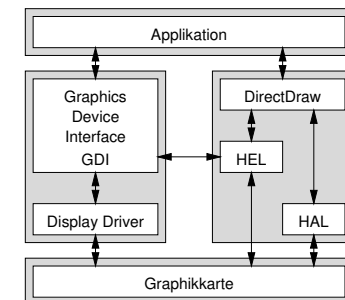
- DirectDraw high-performance 2D-Graphik
- Direct3D 3D-Graphik
- DirectShow Videowiedergabe
- DirectSound grundlegende Audiofunktionen (3D)
- DirectMusic Software-Synthesizer
- DirectInput schnelle Treiber für Tastatur/Maus/Joystick/...
- DirectSetup einfache und sichere Installation
- DirectPlay Multiplayer / Internet
- was fehlt? Sprachein-/ausgabe
...

PC-Technologie | SS 2001 | 18.214

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 Abstraction Layer
- DDI: GDI Device Driver Interface



PC-Technologie | SS 2001 | 18.214

DirectX: Direct3D

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

[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)

PC-Technologie | SS 2001 | 18.214

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)

PC-Technologie | SS 2001 | 18.214

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
- siehe Audio-Folien

PC-Technologie | SS 2001 | 18.214

DirectX: DirectMusic

"message-based musical data"

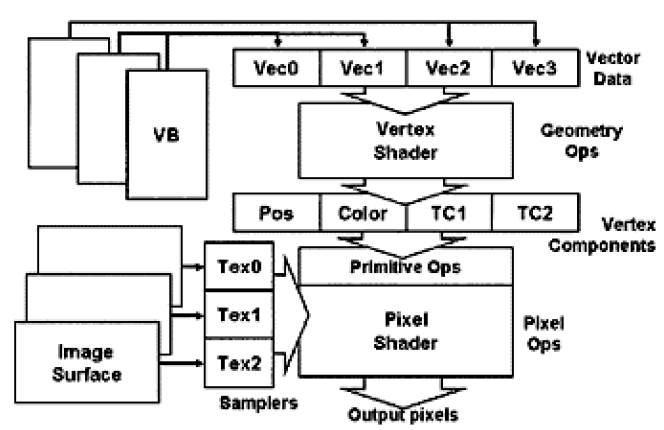
- Noten, Akkorde, Phrasierung, ...
- MIDI-Schnittstelle
- Klangerzeugung über Software-Synthesizer
- und Wiedergabe über DirectSound
- oder Ansteuerung externer Synthesizer
- DLS (downloadable sounds / SoundFont2)
- DirectMusic Producer
- Hardwareunterstützung nur in Windows98/2000

PC-Technologie | SS 2001 | 18.214

DirectX: DirectInput

- Treiberschicht zu Eingabe-Geräten
- Treiber für sehr viele Geräte
- umgeht die normalen Windows-Treiber
- minimale Latenz
- Unterstützung von force-feedback Geräten
- diverse Kraft

DirectX8: Graphics



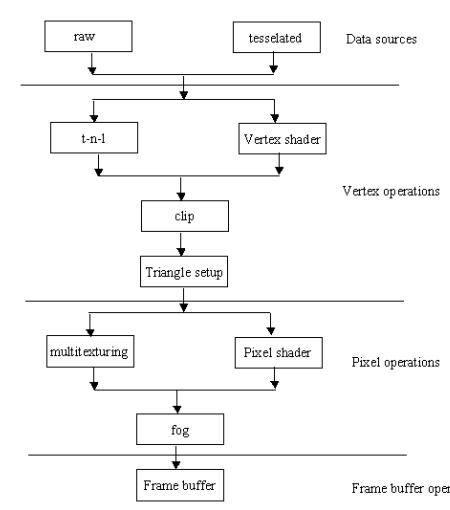
- kombiniert bisherige APIs: DirectDraw und Direct3D
- interne Programmiersprachen für "Vertex/Pixel-Shader"

DirectX 8: VertexShader



- morphing / twining animation
- matrix palette skinning
- user-defined lighting models
- general environment mapping
- procedural geometry
- developer defined algorithms

DirectX 8: Graphik-Pipeline

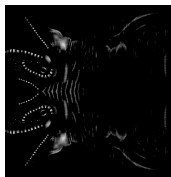


- kompatibel mit DirectX 7
- neue Fkt. nur auf "neuer" HW

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



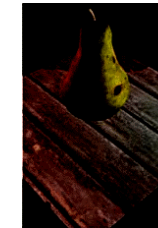
DirectX 8: VertexShader Beispiel...

- Folie noch nicht fertig, siehe MS DX8 Overview (PowerPoint)

DirectX 8: Multitexturing

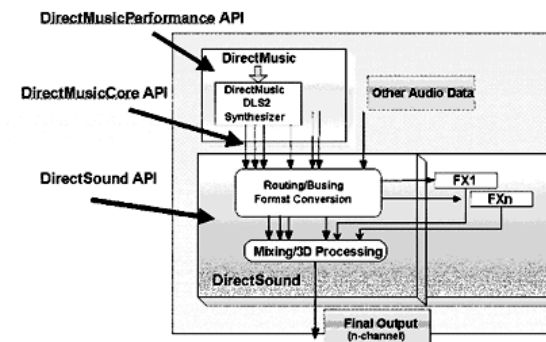


- Überlagerung mehrerer Texturen
- z.B. für light-maps (siehe oben)
- oder bump-mapping (rechts)



Ati Radeon: 2 Pipes, je 3 Texturen
 Geforce 2: 4 Pipes, je 2 Texturen

DirectX 8: Audio

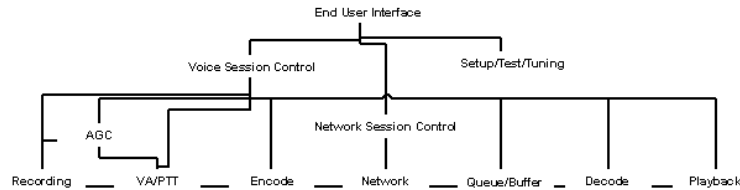


Integration von DirectSound und DirectMusic:

- 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: 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( lpDDSBBack->GetDC( &hdc )) { ... }
if (SUCCEEDED( lpDDSBBack->GetDC( &hdc )) { ... }
```

SDL: "Simple DirectMedia Layer"



Leerseite

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);
}

```

Leerseite