

# Architektur und Entwicklung von VR-Anwendungen (VR-Frameworks)

LMU München  
LFE Medieninformatik  
HS Virtual- und Augmented Reality  
Helge Groß  
SS 2004

## Überblick

- Wofür VR-Anwendungen?
- Architektur von VR-Anwendungen
- Entwicklungsphasen und Workflow
- Entwicklungsumgebungen (IDEs)
- VR-Frameworks und VR-APIs
- Ausblick und Diskussion

# Wofür VR-Anwendungen?

- ▶ Frühe Präsentation und Untersuchung von Produkten ohne physisches Modell
  - ▶ Produkt-Design
  - ▶ Crash-Tests
  - ▶ Material-Tests
- ▶ Durchführung von kinematischen, ergonomischen und aerodynamischen Analysen im Rechner
  - ▶ Automobilindustrie
  - ▶ Luft- und Raumfahrt
- ▶ Für Simulationen, Training, Lehre und Unterhaltung
  - ▶ Computerspiele
  - ▶ VR Sicherheitstraining für gefährliche Gebäude / Arbeitsplätze...

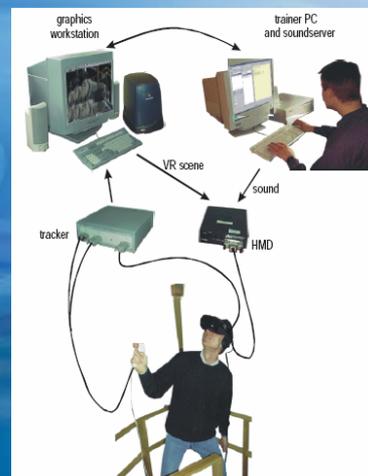
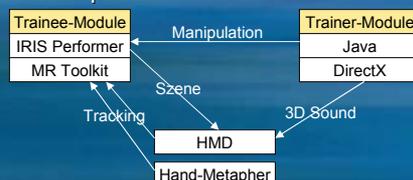
22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

3

# Architektur von VR-Anwendungen I

- ▶ SAVE (Safety Virtual Environment)
  - ▶ Entwickelt für Raffinerie (OMV)
  - ▶ Scene-Simulator: liefert Scene und handhabt Benutzerinteraktionen
  - ▶ Trainer-Application: steuert die zu trainierenden Variablen der Szene (Werte von Instrumenten)
  - ▶ Neue Szenen über UI erstellbar
  - ▶ HMD und Tracking => hoher Immersionsgrad
  - ▶ Komponenten:



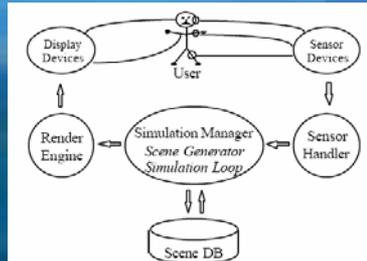
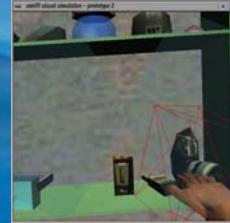
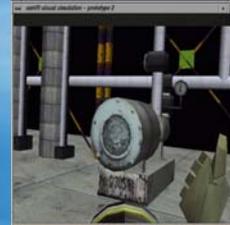
22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

4

# Architektur von VR-Anwendungen II

- Entwickelt als objektorientiertes Design
  - Flexibilität, Verständlichkeit, Erweiterbarkeit
  - Objektorientierung erlaubt VR-Probleme auf sehr hoher Abstraktionsebene zu spezifizieren
- VR-Anwendung als verteilte Architektur (DIVE)
- An Architektur / Dateistruktur einer VR-Engine gebunden



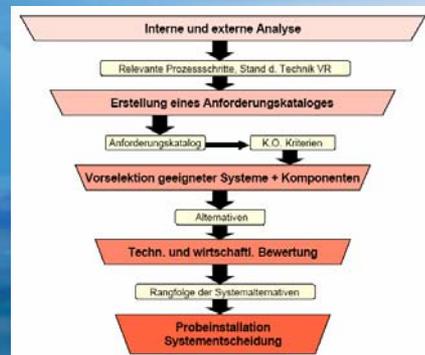
22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

5

# Entwicklungsphasen und Workflow I

- Analyse
  - Analyse des durch VR zu lösenden Problems
  - Finden der Aufgabenstellung
  - Wirtschaftlich
    - Budget
    - Entwicklungszeitraum
- Anforderungskatalog
  - Immersionsgrad
  - Interaktionsmöglichkeiten
  - VR-Hardware Unterstützung (I/O, HMD, Eingabegerät)
  - Performanz (Schnelligkeit)
  - Flexibilität (Portabilität)
  - Desktopanwendung (3D-TFT) oder Gruppenpräsentation (CAVE, Holobench)



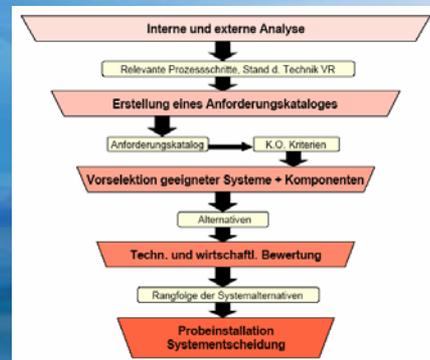
22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

6

## Entwicklungsphasen und Workflow II

- Multiple Pipe-Support (bei Holobench und CAVE: Darstellung auf mehrere Projektionsflächen aufteilen)
- Stereofähigkeit (Bild li. u. re. Getrennt)
- (Head-)Tracking => ständig neue Bildberechnung
- Wahl der geeigneten Komponenten
  - Frameworks, APIs, ...
- Entwicklung alternativer Lösungen
  - Entscheidung zur Entwicklung der auf Anforderungen/Budget passenden



22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

7

## Entwicklungsphasen und Workflow III

- Bei Entwicklung grosser VR-Anwendungen Trennung verschiedener Entwickler-Rollen
- Programmierer
  - Zuständig für Logik und Animation der Anwendung
  - VR ist Echtzeit-3D, kommt ohne Programmcode nicht aus
  - (C++, VR-Frameworks/APIs, VR-Engine-Scriptsprache)
- Modellierer
  - Erstellt die verwendeten 3D-Modelle (3D Studio MAX, Maya)
  - Genaues Vorgehen erforderlich, sonst Grafikfehler in Anwendung
  - Legt Texturkoordinaten fest (UVW), evtl. auch kleine Animationen
- Texturierer
  - Erstellt Bitmaps die auf die Modelle gelegt werden (Materialien)
  - Einzige Vorgabe ist Seitenverhältnis, Mapping von Modellierer oder Programmierer
- Sound-Entwickler
  - Erstellt realistische Geräusche (Audition, OpenAL, PortAudio)

22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

8

# Entwicklungsumgebungen (IDEs) I

- Integrated Development Environment für VR-Anwendungen
  - Anforderungen
    - Rapid Application Prototyping (z.B. ohne Eingabegeräte testen, sondern über Tastatur simuliert)
    - Verständliches GUI
    - Einfach/schnell erlernbar/nutzbar/konfigurierbar
    - Manipulationssprache/Skriptsprache => Modellen unkompliziert Verhalten zuweisen
  - Pro
    - Modelle importierbar (alle gängigen File-Formate)
    - Unterstützen verschiedene Plattformen
    - Unterstützen VR-Geräte
    - Rapid Development navigierbarer 3D-Szenen
    - Libraries mit vordefinierten Objektverhalten
    - Hohe Performanz mit kommerzieller Hardware
    - Mehrbenutzerfähige Anwendungen
  - Contra
    - Spezielle Interaktionen schwierig oder unmöglich zu implementieren

22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

9

# Entwicklungsumgebungen (IDEs) II

- Alice (Rapid Prototyping System for Virtual Reality)
  - Entwickelt seit 1998 an der Carnegie Mellon University (Pittsburgh, PA)
  - Plattform
    - Frei erhältlich, nur Windows
  - Unterstütze VR-Hardware
    - Normale Version nur Keyboard / Maus
    - Erweiterbar für HMD / Datagloves / usw.
  - Beschreibung
    - High-Level Anwendung
    - Einfache Handhabung, ohne technischen Hintergrund erlernbar
  - Manipulationssprache
    - Python: High-Level, interpretiert, objekt-orientiert
    - Erlaubt Unerfahrenen einfach Alice Skripte zu schreiben
  - Einschränkungen
    - Für VR-Geräte wird die Developer-Version benötigt
    - Keine volle Kontrolle über Polygone / Vertices von Geometrien
    - Schlecht bei grossem Datenvolumen und Update für jeden Frame

22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

10

## Entwicklungsumgebungen (IDEs) III

- WorldUp (3D Real-Time GUI)
  - Entwickelt von Sense8, aktuell Version 5
  - Plattform
    - Kommerzielle Software, Windows, UNIX, SGI
  - Unterstützte VR-Hardware
    - Eingabe: Space-Mouse, Space-Ball, Glove, andere
    - Ausgabe: diverse HMDs, Stereo Display, Sound
    - Tracking: High-End-Trachker, Low-End-Tracker
  - Beschreibung
    - High-Level Anwendung
    - Integrierter Modellierer, auch Import: VRML, 3DS, DXF, FLT
    - Parameter des Designs in echtzeit änderbar => schnelle Ergebnisse
    - Browser Plugins zur Verbreitung der erzeugten VR über WWW
    - Verbreitetste VR-Simulations Software
  - Manipulationssprache
    - VB ähnliche Scriptsprache (auch OpenGL und C)
    - Erlaubt schnellstmögliche / sofortige Änderungen der Szene

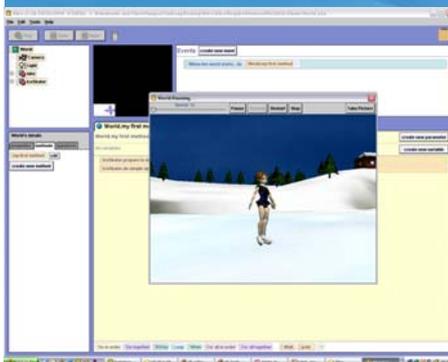
22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

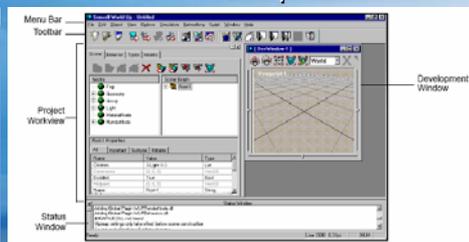
11

## Entwicklungsumgebungen (IDEs) IV

Alice



WorldUp



22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

12

# VR-Frameworks und VR-APIs I

## ➤ Unterschied zwischen API und Framework

- API (Application Programming Interface)
  - Schnittstelle von BS oder Anwendung über die standardisierte Programmbausteine wie Methoden zur Verfügung gestellt werden
  - Erleichtert durch Verbergen der unterliegenden Komplexität die Arbeit des Programmierers
  - Schnittstelle garantiert portabilität des Programmcodes (Implementierung für verschiedene Plattformen)
- Framework
  - Programmgerüst in das an bestimmten Stellen Anwendungsspezifischer Code eingehängt wird
  - Art Standard-Softwarearchitektur
  - Hauptprogramm steckt im Framework und garantiert gemeinsames Verhalten aller daraus entwickelter Anwendungen
  - Eigentliche Applikation hat kein Hauptprogramm mehr und wird von Framework aufgerufen

22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

13

# VR-Frameworks und VR-APIs II

## ➤ Klassifizierung von VR-Frameworks (APIs)

- Grafiksoftware lässt sich in ein Schichtenmodell gliedern
- High-Level
  - Funktionalität zur Unterstützung von Modellieren, Interagieren, Rendern
  - Evtl. Scriptsprache, Grafisches UI, Drag&Drop
  - System übernimmt Berechnungen und Verwaltung der Geometrien
- Low-Level
  - Beste Performanz und Flexibilität erreichbar
  - Viel mehr Operationen => mehr Einfluss und Möglichkeiten
  - Viel komplexer und Kenntnisse in Linearer Algebra zwingend!
- Hardwarenahe Schicht
  - Kommuniziert direkt mit der Hardware

Schicht	Beispiel
Anwendungen (High-Level)	3D Studio MAX, Maya, WorldUp, Alice, ...
High-Level (API/Framework)	Java3D, OpenInventor, WorldToolKit, VR Juggler, dVISE, IRIS Performer
Low-Level (API/Framework)	OpenGL, Direct3D, CAVE, MR Toolkit
Hardwarenah	DirectDraw (Windows)
Hardware	nVidia, GeForce, ATI, ...

22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

14

## VR-Frameworks und VR-APIs III

### ➤ Direct3D

#### ➤ Fakten

- Kommerzielle API von Microsoft für Windows
- Komponente aus Windows DirectX (Multimedia)
- Baut auf Windows 2D-API DirectDraw auf

#### ➤ Unterstützte VR-Hardware

- Nicht direkt, weitere Komponenten nötig

#### ➤ Beschreibung

- Immediate Mode (Direct3D IM)
  - Low-Level, komplex
  - Mehr Freiheiten und Möglichkeiten in 3D-Grafik
- Retained Mode (Direct3D RM)
  - High-Level
  - Setzt auf IM auf
  - Hierarchisches Szenegraph Interface
  - Übernimmt viel Geometrie-Management

## VR-Frameworks und VR-APIs IV

### ➤ MR Toolkit (Minimal Reality Toolkit)

#### ➤ Fakten

- University of Alberta Computer Graphics Research Group (CA)
- Frei verfügbar für Forschung und Wissenschaft
- Plattformen: UNIX, HP, SGI, IBM, (Windows: MRObjets in C++)
- Dient als Grundlage einiger High-Level Tools

#### ➤ Unterstützte VR-Hardware

- Eingabe: Space Ball, 3D Maus, Gloves, ...
- Ausgabe: verschiedene HMDs
- Tracker: Ascension Technologies, Polhemus

#### ➤ Beschreibung

- Low-Level Toolkit mit C, C++ und FORTRAN verwenden
- Keine nutzbare Kollisionserkennung / Mehrbenutzerfähigkeit
- Ermöglicht hardwarenahe VE-Entwicklung
- Keine Unterstützung projektionsbasierter VR-Ausgabe

# VR-Frameworks und VR-APIs V

## WorldToolKit (WTK)

### Fakten

- Das wohl meistbenutzte VR-Framework
- Nicht in jeder Hinsicht das beste, aber das umfassendste
- Kommerzielles VR-Entwicklungs Tool von Sense8
- Plattformen: crossplattform-portabel, SGI, Sun, Windows, Linux

### Unterstützte VR-Hardware

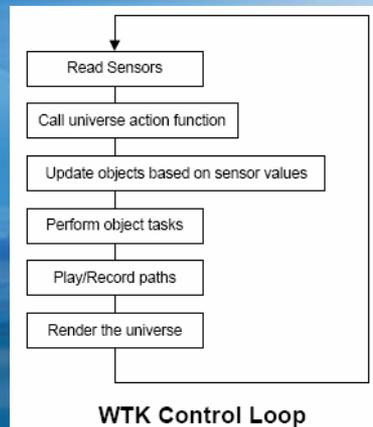
- Grosse Zahl an VR-Ein-/Ausgabe-Geräten wird unterstützt
- Aktuelle Liste: <http://www.sense8.com/products/index.html>

### Beschreibung

- Geschrieben in C, Verwendung der WTK API aus C/C++
- Entwickler muss sich nur um Manipulation der Szene kümmern
- Funktionen auf 20 Klassen verteilt: z.B. Kollisionserkennung
- Import vieler gängiger 3D-File-Formate
- Umfasst auch eine Sound API

# VR-Frameworks und VR-APIs VI

- Mit Hinzunahme von World2World Multiuserfähig (WTK-Server)
- SGI-Version unterstützt Multi-Pipe => nutzbar für CAVEs usw.
- VE ist umfasst vom sogenannten WTK-Universe
- Auf Universe wird der WTK Control Loop ausgeführt



# VR-Frameworks und VR-APIs VII

- VR Juggler (Iowa State University)
  - Fakten
    - Iowa Center for Emerging Manufacturing Technology (ICEMT)
    - Für Wissenschaft und Forschung frei erhältlich
    - Plattformen: SGI, HP-UX, Windows
  - Unterstützte VR-Hardware
    - Ausgelegt auf Projektion (ICEMTs C2), aber auch HMD-fähig
    - 3D-Eingabe- und Trackingunterstützung
  - Beschreibung
    - Entwickler sollen sich auf VR-Welten konzentrieren anstatt auf das System auf dem diese laufen werden
    - Grundlage C++, Verwendung aus C++
    - Ziel: Lösung für alle Arten von VR-Anwendungen
    - Unterteilt in Komponenten für jede Aufgabe, die sog. „Managers“
    - Kernel vereint alle Manager, übernimmt die Kommunikation
    - Eingebautes Performance Monitoring (Time-Stamp Verfahren)

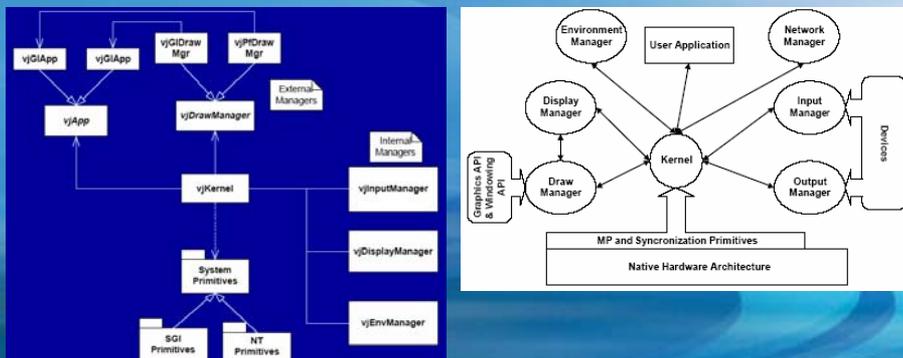
22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

19

# VR-Frameworks und VR-APIs VIII

- Komponenten-Architektur des VR Juggler Frameworks



22.06.2004

Helge Groß - Architektur und Entwicklung von VR-Anwendungen

20

# Ausblick und Diskussion

- ▶ Was glaubt Ihr wie zukünftig VR-Entwicklung aussieht?
  - ▶ Umfassendere GUI-Entwicklungsumgebungen?
  - ▶ Transparentere und einfacher zu erlernende APIs/Frameworks?
  - ▶ Tangible User Interfaces zur Erstellung von VE?
  - ▶ ...?

# Literaturverzeichnis I

- ▶ Wofür VR-Anwendungen?
  - ▶ Institut für Rechneranwendung in Planung und Konstruktion (RPK) (TU Karlsruhe)
- ▶ Architektur von VR-Anwendungen
  - ▶ <http://www.cs.unm.edu/~kuser/TGSD12/HT06/lectures/software.pdf>
  - ▶ <http://www.acorn.uq.edu.au/pdfs/hackmader/vrn.pdf>
  - ▶ [http://www.3.uu.se/edu/course/homepage/isp/933/lectures/isp\\_01\\_intro\\_vr.pdf](http://www.3.uu.se/edu/course/homepage/isp/933/lectures/isp_01_intro_vr.pdf)
  - ▶ <http://webster.fh-hagenberg.at/staff/haller/research/publications/1998/ed98.pdf>
  - ▶ <http://webster.fh-hagenberg.at/staff/haller/research/publications/2001/haller@web3d2001.pdf>
- ▶ Entwicklungsphasen und Workflow
  - ▶ Institut für Rechneranwendung in Planung und Konstruktion (RPK) (TU Karlsruhe)
  - ▶ <http://www.vrjuggler.org/pub/vr.dev.tools.1998.SIGGRAPH98.pdf>
  - ▶ <http://www.lrz-muenchen.de/services/peripherie/vr/vrsoftware.html>
  - ▶ <http://www.medien.informatik.uni-muenchen.de/lehre/ws0304/iv3d/material.html>
- ▶ Entwicklungsumgebungen (IDEs)
  - ▶ <http://www.diverse.vt.edu/VRSsoftware/ist.html>
  - ▶ <http://www.vrjuggler.org/pub/vr.dev.tools.1998.SIGGRAPH98.pdf>
  - ▶ <http://www.sense8.com/products/index.html>
  - ▶ <http://www.alice.org>
  - ▶ <http://www.cs.unt.edu/~steiner/5330-VR/lecture2.html>

# Literaturverzeichnis II

## VR-Frameworks

- <http://www.digens.vt.edu/VRSoftware.html>
- [http://www.vrjuggler.org/pub/vr\\_dev\\_tools\\_1998\\_SIGGRAPH98.pdf](http://www.vrjuggler.org/pub/vr_dev_tools_1998_SIGGRAPH98.pdf)
- <http://www.sense8.com/products/index.html>
- <http://math.mcgill.ca/~math/mcg/courses/03/07/Dirvd030/default.aspx>
- <http://www.cs.wpi.edu/~mat/courses/cs563/talks/cbvrdr/pres3.html>
- [http://www.vrjuggler.org/pub/vr\\_juggler\\_framework\\_for\\_vr\\_dev\\_1998 IPT98.pdf](http://www.vrjuggler.org/pub/vr_juggler_framework_for_vr_dev_1998 IPT98.pdf)
- [http://www.cymt.aus.dk/education/teaching/e03/cvg9/vr/vr\\_juggler\\_programmer\\_guide\\_pages.pdf](http://www.cymt.aus.dk/education/teaching/e03/cvg9/vr/vr_juggler_programmer_guide_pages.pdf)
- <http://www.vrjuggler.org/pub/vrjuggler-afaa2002.pdf>
- [http://www.vrjuggler.org/pub/ieeeVR\\_2001\\_vrjuggler\\_slides.pdf](http://www.vrjuggler.org/pub/ieeeVR_2001_vrjuggler_slides.pdf)
- <http://de.wikipedia.org/wiki/Framework>
- <http://de.wikipedia.org/wiki/API>
- <http://www.vrjuggler.org/about.php>
- <http://www.cs.ualberta.ca/~graphics/MRToolkit/>
- [http://cggmwww.csie.nctu.edu.tw/courses/vr/2001/doc/relative/Intro\\_to\\_WTK.ppt](http://cggmwww.csie.nctu.edu.tw/courses/vr/2001/doc/relative/Intro_to_WTK.ppt)
- <http://www.ifi.unizh.ch/~noser/COURSES/cgVorlesung4.pdf>

**Danke für Eure Aufmerksamkeit!**