



GammaBlitzBoys (100)

InstantReality
- in der AG WBS -
Stand Oktober 2010

cocoon-inga (4)

(18)

(2)

crime82 (8)

The Good Left Ungaga

T.S.R.

New York

romm (16)

Rice and Bread

floxe (6)

Soerinho21 (3)

datze (8)

MissesEff (29)

KuhTin (8)

steteffi (7)

(8)

patrone (9)

muxe (17)

spitout (5)

Niblonian (5)

tes_schaaf (29)

docsnyder1984 (1)

crimeFM (7)

millerdnd (8)

(6)

festivalkult (100)

Sancho Garcia (9)

EisFrei (8)

Überblick

- ▶ **Arbeitsumgebung WBS**
- ▶ **InstantReality**
 - ▶ Starten
 - ▶ Arbeiten mit mehreren Projekten
 - ▶ `dvlplInstantReality.template`
- ▶ **Basisprojekte**
 - ▶ `irBasics.pack`
 - ▶ `irioARTpro.lib`
 - ▶ `irioEyeTracker.lib`
 - ▶ `irioInteraction.pack`

Arbeitsumgebung WBS

- ▶ Einrichten der Arbeitsumgebung im WBS Netz
 - ▶ Registrierung im WBS Wiki unter
 - ▶ <https://hiro.techfak.uni-bielefeld.de/twiki/bin/view/>
 - ▶ Auth-Login: agwbs
 - ▶ Wichtig
 - TechFak E-Mail Adresse verwenden!
 - Login Name:VornameNachname
 - Ansprechpartner: Nikita Mattar (nmattar@t...)
 - ▶ Erste Anlaufstelle
 - ▶ Arbeitsumgebung einrichten
 - ▶ <https://hiro.techfak.uni-bielefeld.de/twiki/bin/view/Entwicklung/ArbeitsumgebungEinrichten>
 - ▶ Entwicklungsumgebung initialisieren
 - ▶ <https://hiro.techfak.uni-bielefeld.de/twiki/bin/view/Entwicklung/WVSSkript>

Starten von Instant Reality

- ▶ **Voraussetzungen**
 - ▶ InstantReality im Pfad aufnehmen
 - ▶ wbs_rcinfo verwenden
 - export WBS_RCINFO_ILIST="local global ir opensg system"
 - ▶ Java-Classpath korrekt gesetzt
 - ▶ LD_LIBRARY_PATH korrekt gesetzt
 - ▶ wichtig: OpenSG später im Pfad für InstantPlayer
 - ▶ OpenSG früher im Pfad für alte OpenSG Projekte (BioSphere)
 - ▶ **InstantPlayer** eingeben
- ▶ **Einfacher: irp Skript**
 - ▶ Liegt im irBasics.pack, ist aber auch global installiert

Das Skript irp

- ▶ Einfacher Aufruf am Desktop:
 - ▶ `irp Szene.x3d`
- ▶ Einfacher
 - ▶ im Projektverzeichnis `data/x3d` ein `main.x3d` ablegen
 - ▶ dann `irp` irgendwo im Projektverzeichnis starten
 - ▶ `irp` findet die Datei dann automatisch
- ▶ Desktop vs. CAVE vs. Panel
 - ▶ **`irp -d`** startet am Desktop
 - ▶ **`irp -c`** startet in der CAVE
 - ▶ **`irp -w`** startet auf den drei Panels
 - ▶ Weiteres offenbart **`irp --help`**

Wie wird das gemacht?

▶ **irp -c**

- ▶ Lokalisiert eine Datei CaveSetup.x3d
- ▶ Injiziert die eigentliche Startdatei in die CaveSetup.x3d Datei (per inline)
- ▶ Startet modifizierte CaveSetup.x3d Datei
- ▶ Dadurch übernimmt irp die Konfiguration der
 - ▶ Displays und der
 - ▶ Tracker

Arbeit mit mehreren Projekten

▶ Problem

- ▶ Dateipfade in x3d Dateien müssen aufgelöst werden, einen dynamischen Mechanismus wie bei LD_LIBRARY_PATH gibt es nicht
- ▶ Pfade müssten also vorher bekannt sein
- ▶ Schwierig bei Mehrbenutzer-Betrieb mit flexiblen Verzeichnisstrukturen

▶ Lösung

- ▶ irp startet einen Modifying Proxy
 - ▶ Kann als Quelle für Dateien angegeben werden (images, inlines, volumes, sounds)
 - `<Inline url=http://localhost:43093/emma/emma.wrl/>`
 - ▶ Ist aber umständlich (wie ist denn jetzt meine Portnummer?), daher macht irp das schon automatisch! ;-)
 - `<Inline url="emma/emma.wrl"/>`
 - ▶ Alle weiteren Zugriffe laufen damit immer über den Proxy
 - ▶ irp sucht die Dateien in Pfaden, die angegeben werden in der Variable
 - `IRP_DATA_PATH=„./vol/wbsopt/data/models:/vol/wbsopt/data/x3d“`
 - ▶ und in einer evtl. projektspezifischen Datei „config/irp.default.sh“ oder „config/irp.PROJECTNAME.sh“

Ein neues Projekt anlegen

- ▶ `wvs create PROJECTNAME InstantReality`
- ▶ Erzeugt Standard-Dateibaum mit
 - ▶ ant build script für JAVA-Knoten
 - ▶ Settings-Datei für irp
 - ▶ `data/x3d/main.x3d`



Projekte

irBasics.pack

- ▶ **Konfigurationsdateien (x3d) für**
 - ▶ Cave, Desktop, Panels
 - ▶ ART Tracking
- ▶ **Script4X3D**
 - ▶ ermöglicht Online-Scripting
 - ▶ Einfache Editierfunktion für Objekte
 - ▶ Szenengraph-Manipulationen
- ▶ **Model der CAVE**
 - ▶ nützlich zur Planung des Interaktionsraumes

irioARTpro.lib

- ▶ Anbindung des ART Tracking Systems
 - ▶ Korrigierte Rotationen
- ▶ Anbindung der Handschuhe
 - ▶ Tracking
 - ▶ Feedback

irioEyeTracker.lib

- ▶ **Anbindung des Arrington Research Eyetrackers**
 - ▶ Windows Server über InstantIO
 - ▶ Client als InstantIO Import
 - ▶ Kalibrierung über X3D/Javascript/Java
- ▶ **Beispiel-Dateien im Projekt**

irioInteraction.pack

- ▶ Debug-Viewer für InstantIO Updates

Dokumentation WBS

- ▶ Wiki

- ▶ <https://hiro.techfak.uni-bielefeld.de/twiki/bin/view/Entwicklung/InstantReality>