



Virtuelle Realität und Simulation - Übung 1

Einführung

Prof. Dr. Gabriel Zachmann (zach@in.tu-clausthal.de)

David Mainzer (dm@tu-clausthal.de)

Institut für Informatik

7. November 2008



Gliederung

Allgemeine Informationen

Übungsablauf

Software – Viewer/Editor

VRML-Viewer

Editor

Programmiersprachen – EAI Manipulation

Javascript

Java

Debugging

Scenegraph

Ihre Aufgabe



Gliederung

Allgemeine Informationen

Übungsablauf

Software – Viewer/Editor

VRML-Viewer

Editor

Programmiersprachen – EAI Manipulation

Javascript

Java

Debugging

Scenegraph

Ihre Aufgabe



Übungsablauf

Zeiten

- Alle 2 Wochen Freitags im IFI Raum 202 (Powerwall)
- Übungsblatt erscheint alle 2 Wochen

Übungen

- Vorstellen des **neuen** Übungsblattes
- Abgabe des letzten Übungsblattes (inkl. kurzer Erläuterung)



Gliederung

Allgemeine Informationen

Übungsablauf

Software – Viewer/Editor

VRML-Viewer

Editor

Programmiersprachen – EAI Manipulation

Javascript

Java

Debugging

Scenegraph

Ihre Aufgabe



VRML-Viewer

- Auf der [VL-Website](#) finden Sie Links zu diversen Viewern
 - *InstantReality* (Windows / OSX / Linux)
 - *FreeWRL* (Linux / OSX (SourceCode verfügbar))
 - *Cortona* dies ist ein kommerzielles Plugin für Browser
- Rate zu *InstantReality* (bisher kein ebuild für Gentoo (amd64))
 - Entwickelt durch Fraunhofer IGD
 - Besitzt eine Art Debug-Konsole (siehe Anhang)
 - Unterstützt Java und .Net für EAI (External Authoring Interface) Manipulation



Editor

- Hier sollte jeder seinen eigenen Favoriten wählen (rein Text basiert oder mit grafischer Oberfläche)
- *X3D-Edit* scheint ein recht ausgereifter Editor zu sein
- Auch *Notepad / Kwrite ...*
- Selber nutze ich *vim / gvim* (existiert Syntax-Highlighting)



Gliederung

Allgemeine Informationen

Übungsablauf

Software – Viewer/Editor

VRML-Viewer

Editor

Programmiersprachen – EAI Manipulation

Javascript

Java

Debugging

Scenegraph

Ihre Aufgabe



Javascript – ECMAScript (ECMA 262)

- Häufig für DOM Scripting in Browsern eingesetzt
- Wird zur Laufzeit interpretiert
- Von allen aktuellen Browsern unterstützt – eventuell aus Sicherheitsgründen deaktiviert



Java



- **Java** – weitverbreitet objektorientierte Programmiersprache (sollte jedem bekannt sein)
- Entwickelt von Sun Microsystems
- Muss vorab in ByteCode übersetzt werden
- Von *InstantReality* unterstützt

Hello World – Vorlage

```
# VRML V2.0 utf8

Background { skyColor 0 0 1 }

5 Transform {
  translation 0 0 0
  children [
    DEF SCHRIFT Transform {
      children [
        10 Transform {
          scale 0.2 0.2 0.2
          translation 0.2 0 0
          children [
            15 Inline { url "penguin.wrl" }
          ]
        }
        20 Shape {
          geometry Text { string ["Hello_World"] }
        }
      ]
    }
  ]
}
```

Hello World – Abschnitt 2 in Java

```
20 DEF ROTOTATOR Script {  
    eventIn      SFFloat      set_fraction  
    eventOut     SFRotation   value_changed  
    url          "hello_java.class"  
}  
  
25 DEF TIMER TimeSensor {  
    cycleInterval 10  
    loop TRUE  
}  
  
30 ROUTE TIMER.fraction_changed TO ROTOTATOR.set_fraction  
ROUTE ROTOTATOR.value_changed TO SCHRIFT.set_rotation
```

- Wichtig:
 - url hier **hello_java.class**
 - **eventIn** und **eventOut**

Hello World – JavaFile hello_java.java

```
import vrml.field.*;
import vrml.node.*;
import vrml.*;

5 public class hello_java extends vrml.node.Script {
    float[] rotation=new float[4];

    public void processEvent(Event e) {
        if (e.getName().equals("set_fraction")) {
10 //System.out.println("silly debug message");
            rotation[0]=0.0f;
            rotation[1]=1.0f;
            rotation[2]=0.0f;
            rotation[3]=((ConstSFFloat)e.getValue()).getValue()
15 *2.0f*(float)Math.PI;
            SFRotation value_changed=(SFRotation)getEventOut("value_changed");
            value_changed.setValue(rotation);
        }
    }
20 }
```

- Kompilieren einer *.java Datei (benötigt **jdk**):
 - **javac -classpath PATH/TO/instantreality.jar hello_java.java**
(default: INSTALL/InstantPlayer/bin)

Hello World – Abschnitt 2 in JavaScript

```
20 DEF ROTOTATOR Script {  
    eventIn      SFFloat      set_fraction  
    eventOut     SFRotation   value_changed  
    url          "hello_javascript.js"  
}  
  
25 DEF TIMER TimeSensor {  
    cycleInterval 10  
    loop TRUE  
}  
  
30 ROUTE TIMER.fraction_changed TO ROTOTATOR.set_fraction  
ROUTE ROTOTATOR.value_changed TO SCHRIFT.set_rotation
```

- Wichtig:
 - url hier `hello_javascript.js`
 - `eventIn` und `eventOut`



Hello World – JavascriptFile hello_javascript.js

```
function set_fraction(value) {  
    value_changed[0]=0;  
    value_changed[1]=1;  
    value_changed[2]=0;  
5    value_changed[3]=value*2.0*Math.PI;  
}
```

- Keine Kompilierung notwendig!
- Kann auch direkt in das *.wrl File geschrieben werden



Gliederung

Allgemeine Informationen

Übungsablauf

Software – Viewer/Editor

VRML-Viewer

Editor

Programmiersprachen – EAI Manipulation

Javascript

Java

Debugging

Scenegraph

Ihre Aufgabe

Debugging – Javascript (benötigt Konsole)

```
DEF Debug Script {  
  eventIn MFVec3f set_coord  
  eventIn SFFloat set_float  
  url [ "javascript:  
5     function set_coord( value, timestamp ) {  
        print( 'Debug: coord = ' + value + '\n' );  
      }  
      function set_float( value, timestamp ) {  
10         print( 'Debug: float = ' + value + '\n' );  
      }"  
  ]  
}
```

- Definiere neuen Knoten Debug
- Verwende ROUTE um Ereignisse bekannt zu geben

```
ROUTE *****.fraction_changed TO Debug.set_float  
ROUTE *****.value_changed TO Debug.set_coord
```

Debugging – Logger (**nur** InstantReality)

```
DEF Log Logger {  
  level 3      # 0 ...  
  logFile ""   # default = console  
}
```

- Definiere neuen Knoten Log
- Verwende ROUTE um Ereignisse an ihn zu senden
- Timestamp wird automatisch hinzugefügt (je nach Log-Level)

```
ROUTE *****.fraction_changed TO Log.write
```

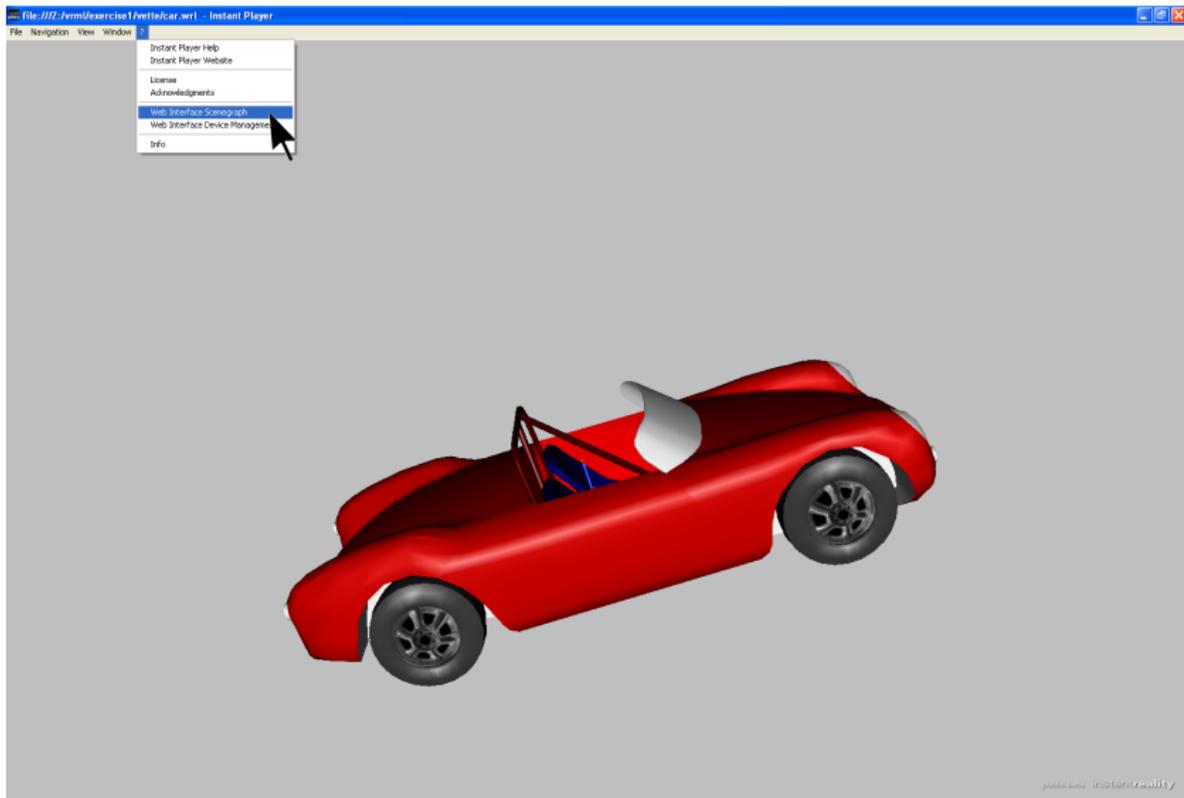


InstantReality – Scenegraph

- InstantReality kann Knoten der Szene anzeigen
- Müssen dafür Definiert werden (Verwenden Sie sinnvolle Bezeichner)

```
ROUTE *****.fraction_changed TO Log.write
```

- Eigenschaften der Knoten können auch verändert werden





Avalon Web Interface - Mozilla Firefox

File Edit View Bookmarks Tools Help

http://localhost:35668/index.html

Avalon Web Interface © 2002 JGD

System:

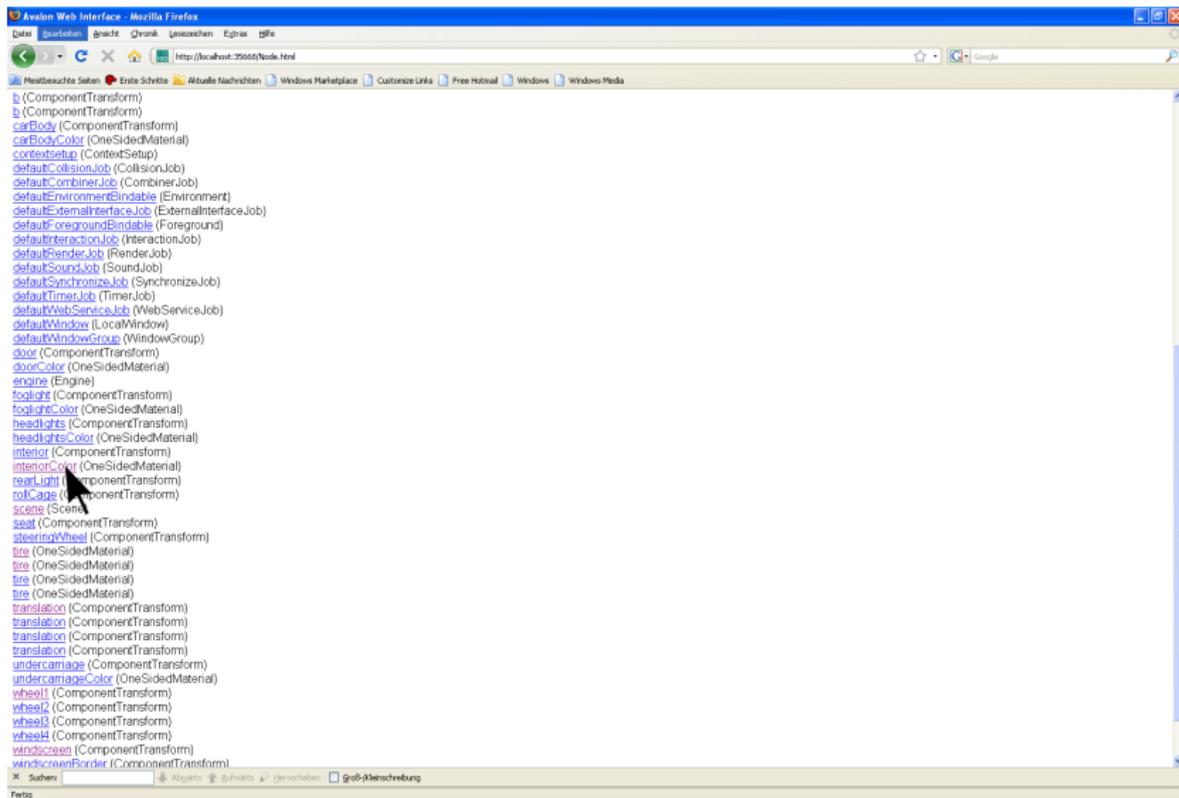
User: dm
 Host: maxwellWindows XP
 HostID: MnVNu+yEALNnrD0zqp/skcZLUXOY=
[Server](http://maxwell.in.tu-clausthal.de:35668): http://maxwell.in.tu-clausthal.de:35668
 Version: V2.0.0beta5 build: R-10580 Aug 7 2008
[Context](file:///Z:/vml/evexercice1/vette/car.wrl): file:///Z:/vml/evexercice1/vette/car.wrl
[Setup](#): 36 dynamic parameter

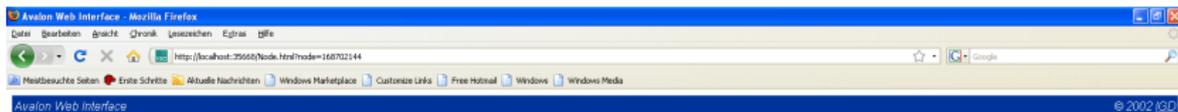
Generic Pages:

Object Type	Count	Page
BaseObject	6182	
Route	27	
Bindable	6	All
Node	225	Named
Namespace	9	All
Node Type	333	All
Field Type	51	
Proto	0	

X Suchen Absolut Relativ Ubersichten Groß-/Kleinschreibung

Fertig





DEF interiorColor OneSidedMaterial

[AN_NS_169622000](#)

ID: 168702144

State: Live

type: OneSidedMaterial

proInfo: corePtr not set

parents: [ManagedAppearance](#)

Fields

name	group	type	data type	value	short cut	D	A	P	A	E	X	T	S
locFeature	EF	MFString			0	1	0	0	0				
ambientIntensity	EF	SFFloat		1.73205	2	1	0	0	0				
diffuseColor	EF	SFColor		1.0 0	3	1	0	0	0				
emissiveColor	EF	SFColor		0.0 0	4	1	0	0	0				
shininess	EF	SFFloat		1	5	1	0	0	0				
specularColor	EF	SFColor		0.0 0	6	1	0	0	0				
transparency	EF	SFFloat		0	7	1	0	0	0				

In Slots

[name](#) [data type](#) [Routes](#) [ID](#) [S](#)

Out Slots

[name](#) [data type](#) [Routes](#) [ID](#) [S](#)

Children

[name](#) [Link](#)

metadata None





Gliederung

Allgemeine Informationen

Übungsablauf

Software – Viewer/Editor

VRML-Viewer

Editor

Programmiersprachen – EAI Manipulation

Javascript

Java

Debugging

Scenegraph

Ihre Aufgabe