



MOTIVATIONS

WHY 3D INTERFACE / VIRTUAL ENVIRONMENT ON THE WEB ?

- Is *“surfing the Net”* a form of VR ?
- Why VRML is a form of VR ?
 - define VR
 - immersive vr. non-immersive VR



MOTIVATIONS

WHY 3D INTERFACE / VIRTUAL ENVIRONMENT ON THE WEB ?

- ***“Without space we are lost”***
 - spatial memory vr. literal memory
- Collective Visualization
 - visualization of abstract information and/or complex data
- ***“I hear, I forget
I see, I remember
I experience, I understand”*** ---Chinese Saying
- new forms of communication
 - collaborative virtual environments



The very beginning;

when all was science-fiction

“ Cyberspace. A consensual hallucination experienced daily by billions of legitimate operators, in every nation, by children being taught mathematical concepts...A graphic representation of data abstracted from the banks of every computer in the human system. Unthinkable complexity. Lines of light ranged in the nonspace of the mind, clusters and constellations of data. Like city lights, receding...”

**from Neuromancer
by William Gibson**



How does VRML Work ?

- VRML uses the existing Web Infrastructure
- Mime Extension: VRML 1.0
VRML 2.0
- VRML needs a browser to display the worlds

QvLib

VRML 2.0 Browsers

- CosmosPlayer
 - Available on Windows95, WindowsNT and Silicon Graphics
 - Enables high-performance interaction with large-scale worlds
 -
- Sony
- Liquid Reality



Creating VRML WORLDS

3D Primer

- Understanding the 3D Space
- Polygons: the building world for a computer generated 3D world
- Lighting: Point light, Directional, Spot light
- Rendering
- Texture Mapping

VRML concepts

Note: VRML is case sensitive

- Nodes
- Fields: parameters of the Nodes
- Linking
- Constructs



Nodes (primitives)

- shape nodes - describe 3D objects such as box, sphere, cylinder, cone, text etc.
- property nodes - specify appearance and qualitative characteristics of objects such as material, lighting model, textures or environment (?)
- group nodes (nested nodes) - act as managers for other (children) nodes and impose certain specific properties for group members
- sensors



CURIOSITIES

What is VRML ?

- **the answer to the navigation problem on The Web**
- **an interpreted language for describing multi-participant interactive simulations; virtual words networked and hyperlinked within the Web**
- **a standard (ISO/IEC 14772) for specifying 3D graphical words on the Web**
- **a subset of Open Inventory**

What does it stand for ?

Now: Virtual Reality Modelling Language
In the beginning: Virtual Reality Markup Language

How is it pronounced ?

“Vermal”



HISTORY

Where did it come from ?

- Founding Fathers: Mark Pesce and Tony Parisi
- Geneva : First International Conference on the World Wide Web, Spring '94

Ragett:

“Extending WWW to Support Platform Independent Virtual Reality”

- Lessons learned from experience with HTML
- Adapt an existing 3D graphics language
- VRML Forum was formed to involve the Web community in the discussion of realising a VR language for the Web
So, the ***www-vrml*** mailing list was created



Evolution

- **Silicon Graphics' Open Inventor:**
 - used for rapid prototyping of 3D applications
 - SG had produced a number of tools
- **QvLib:**
 - a quick VRML parser written in C++
 - was made public domain in Dec. '94 to implement browsers
- **Day Zero:**
 - 3 April '95 when SG introduced WebSpace,
the first VRML browser
- **4 Aug. '96:**
 - Final Specification of VRML 2.0 named Moving Worlds



From VRML 1.0 to Moving Worlds

VRML 1.0:

- a subset of Open Inventor which included hyperlinked nodes
- all interaction was left out

VRML 2.0

- enhanced static worlds
 - introducing new multimedia nodes
- interaction - sensor nodes
- animation
 - pre-define animation objects called Interpolators
- scripting - scripts and events
- prototyping
 - support for self-describing nodes and browser extensions



How does VRML work ?

- **VRML uses the existing Web infrastructure**
- **MIME Extensions:**

VRML 1.0	x-world / x-vrml
VRML 2.0	model / vrml
- **VRML needs a browser to display the worlds**
 - **Plug-Ins or Viewers:** embedded in Netscape 2.0
 - **Stand-Alone Browsers:**
invoke Netscape 2.0 to display HTML documents

VRML 2.0 Browsers

SG` CosmosPlayer (beta version) : configures Netscape automatically to view VRML worlds

- **Available on Windows95: min. requirements**
Pentium, 16 MB RAM, Netscape 2.0
WindowsNT and Silicon Graphics
 - **Enables high performance interaction with large-scale worlds**
 - **Hypermedia**
 - **VRML + Java + HTML + MM = taking off to the next level of the Web**
- **Sony `s CommunityPlace: multi-user shared words**
 - **Dimension -X `s Liquid Reality: Java toolkit with VRML functionalities**

- Real Space's RealVRTraveler : **Plug-In or Stand-Alone for the Macromedia Director**
- Netscape's Live3D



CosmosPlayer Navigation Interface

THE VIEWER MENU

VIEWER : * **WALK -- WALK THROUGH A SCENE**
* **EXAMINE -- EXAMING OBJECTS**

THE DASHBOARD CONTROLS CHANGE DEPANDIING ON THE VIEWER OPTIO YOU SELECT.

VIEWPOINTS: * **NEXT**
* **PREVIOUS**
* **LIST OF ALL AVAILABLE VIEWPOINTS**

TRAVEL SPEED: * **TRAVEL FASTER**
* **TRAVEL AT DEFAULT SPEED**
* **TRAVEL SLOWER**

IMAGE QUALITY: * **BEST QUALITY**
* **SMOOTH MOTION**

HEADLIGHT ON / OFF

SHOW DASHBOARD

JUMP CUTS -- JUMP TO THE VIEWPOINT YOU WANT



Authoring Tools (freely available on the Web)

Caligari's Pioneer for Windows95

Convertors

- from CAD to VRML
- from VRML 1.0 to VRML 2.0



Creating Moving Worlds

3D Primer

- **Understanding the 3D space**
 - Cartesian Right-Handed System
 - Viewing model: viewpoint of the camera
- **Polygons: the building world for a computer generated 3D world**
- **Lighting: objects are visible because they either emit or**
 - **Point light: radiate out equally**
 - **Directional light: has 1) location and 2) orientation (= direction it points)**
 - Spot light: 3) focus
- **Shading: when an object has been lit, it is shaded**
- **Material**
 - reflected
 - specular colour
 - diffuse colour
 - transmitted
 - absorbed
- **Texture Mapping: applying a picture on a surface**



VRML Basics

- **file extension: .wrl**
- **ascii file format**
- **case sensitive**
- **at the highest level of abstraction: objects that can contain anything**

VRML Concepts

- **Nodes: building blocks (objects) of VRML**
they are arranged in hierarchical structures called scene graphs
(: define an ordering in a group of nodes)
- **Fields: parameters of the the Nodes**

Node Characteristics

- **type name**
- **parameters i.e. fields:**
Fields: private
ExposedFields: public i.e. can be modified by other nodes
- **a set of events that nodes can send and receive**
--- Nodes have at least one event: eventIn (= receive events) and some of them have eventOut



Types of Nodes

- **shape nodes** -- describes 3D objects as box, sphere, cylinder, cone, asciitext etc.
- **property nodes**-- specify appearance and qualitative characteristics of objects such as material, lighting model, textures or environment
- **group nodes (nested nodes)** -- act as managers for other (children) nodes and impose certain properties for group members

Example (Primitives)

```
#VRML v2.0 utf8
```

```
Shape {
    appearance NULL
    geometry NUL
}
```

```
Shape {
    geometry Box { }
}
```

```
Box { size 2 2 2 }
```

```
Shape {
    geometry Box {
        size 3 2 1
    }
}
```



```
Sphere { radius .2 }
```

```
Cylinder { radius 1  
           height 2  
           bottom TRUE  
           side TRUE  
           top TRUE  
         }
```

```
Cone { bottomRadius 1  
       height 2  
       bottom TRUE  
       side TRUE  
     }
```

```
appearance Appearance {
```

```
  material Material {  
    diffuseColor .8 .8 .8  
    emissiveColor 0 0 0  
    specularColor 0 0 0  
    shininess .2  
    transparency 0
```

```
  }
```



```

Transform { center 0 0 0
            translation 0 0 0
            rotation 0 0 1 0
            scale 1 1 1
            scaleOrientation 0 0 1 0
            bboxCenter 0 0 0
            bboxSize -1 -1 -1
            children [ .... ]
            }

```

Anchor (grouping node)

- causes a URL to be fetched over the network when the viewer activates an object
- specifying viewpoint

legal vr. non- legal VRML

Example

```

Anchor {
    url "http://foo.com/vrml/world2.wrl"
    children [ ... ]
}

```

or #Overview



Inline

- reads its children data from a location on the Web

Example

```
Inline {  
    url [ ]  
}
```

Sound

3D spatialised Audio for realism eg. phone

- MIDI files for background soundtracks
- .wav files can be triggered randomly

Sensors

- mechanism for the user to interact with objects
- they generate events based on time or user's action (eg. clicking the mouse)

- ProximitySensor
- SphereSensor
- TouchSensor
- VisibilitySensor
- TimeSensor
- PlaneSensor



Script

- **implement behavior ; analyses the input and outputs an event to change the world**
- **can contain: -- source code
-- pointers to source**

Script Languages: no official designated languages

- **JavaScript: .js**
Note: Perl, Tcl, C can also be used as they are VRML 2.0 compliant
but there are no browsers to support them
- **Java: java byte code**
- **Inline Javascript often called VRMLscript**

Example:

```
Script {  
    eventIn SFBool start  
    url [ "http://foo.com/fooBar.class",  
          "http://foo.com/fooBar.js",  
          "javascript:function start (value, timestamp)"  
    ]  
}
```



Example

```
Script {  
    eventIn SFBool start  
    url "http://foo.com/Example.class"  
}
```

Example.java

```
import vrml.*;  
import vrml.field.*;  
import vrml.node.*;  
  
class Example extends Script {  
    .....  
    { public void processEvent (event e)  
    }  
}
```

Interpolators

```
-- provide the animation which is a kind of behavior; built-in  
-- linear keyframe animation: starting with some knowing  
information  
    and using it to estimate some other unknown information
```



Events and Routes

Route: establishing event paths between nodes

Prototyping

- **extends node type**
-
- **re-usable objects**
- **browser extensions**
- **implemented by using the PROTO keyword**

