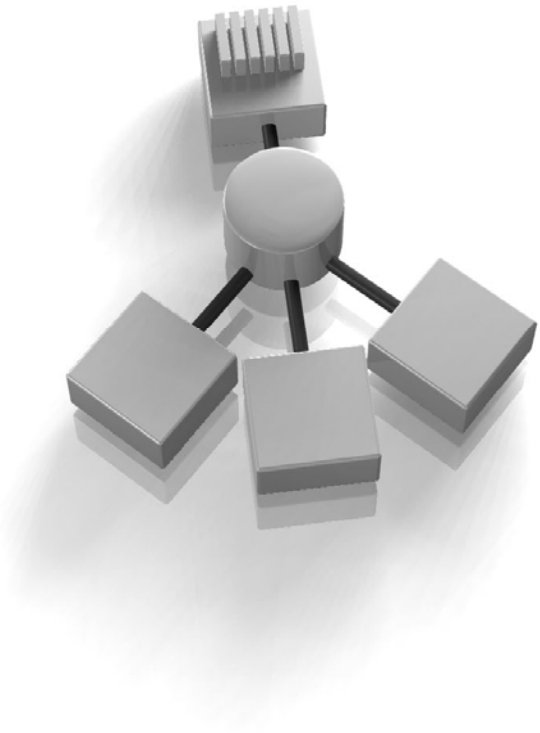
SIXTH FRAMEWORK PROGRAMME
PRIORITY IST-2002-2.3.1.8
Networked Audiovisual Systems

Uni-verse project

Deliverable D 4.1

Simple Rendering Client

Feb 1, 2005



STREP project
Project acronym: Uni-Verse
Project full title: A Distributed Interactive Audio-Visual Virtual Reality System
Proposal/Contract no.: 002228
Distribution: Public

Content

WORKPACKAGE DESCRIPTION		
Workpackage Title: **Rendering Clients development**		WP No: **4**
Starting date: month **.4**	Duration: **24** months	Total Effort in Man-months: **32**
Member involved	Task description / Contribution of Member	Effort man-months
KTH	Participant	**14**
II	Participant	**0**
HUT	Participant	**0**
FHG/IGD	**Workpackage leader, Participant**	**18**
MINUSPLUS	Participant	**0**
PAREGOS	Participant	**0**
BLENDER	Participant	**0**

Objectives
Existing rendering clients will be adapted to various hardware platforms, such as CAVEs and PDAs, complementing existing desktop renderer.

Description of work / tasks
The work will comprise the following tasks:

WP 4.1 High-performance Rendering client
Adaptation and testing of rendering clients for PC, Workstations and Immersive Environments

WP 4.2 Slim Rendering client
Development and testing of rendering client for mobile phones and PDAs based on OpenGL ES.

Deliverables
The deliverables of WP 4 are:

D 4.1 Simple client (first release of High-Performance Rendering client)(Month 12)
D 4.2 High-Performance Rendering clients (Month 24)
D 4.3 Slim Rendering Client (Month 28)

Milestones
M 4.1 Simple client released. (Month 12)
M 4.2 Slim rendering client; works according D 2.3 (Month 28)

Interrelation with other workpackages
This WP will be crucial for supplying WP7 and WP8 with rendering tools for testing and evaluation

Deliverable D4.1 Simple Rendering Client

Summary

This document, D4.1, is the description of the “Simple Rendering Client” which is a prototype of the planned high-performance rendering client in Uni-Verse project, A Distributed Interactive Audio-Visual Virtual Reality System, proposal/Contract no.: 002228.

About the Author

This report was written by Emil Brink, who is employed by partner KTH as a developer/engineer. Emil has a Master of Science in Computer Science and Engineering from KTH, and also worked with Eskil Steenberg on the original Verse implementation at the Interactive Institute in 1999 and onwards.

Description

“Render” is a very simple prototype-quality rendering client for Verse. It was originally written in 1999 for the first incarnation of the Verse system, and ported to the current API in 2004.

Render is intentionally written as a very primitive client, whose main purpose is to get the first “polygon on screen”. Being able to see geometric data transmitted over the Verse protocol is very handy during development and testing of other components, from core protocol code, to the API, to the host implementation.

Render does not have a graphical user interface for managing the connection to a server. Instead, it accepts the address to use as a command-line argument, and immediately tries to connect after starting. Once connected, it will ask the host for all object nodes. As object nodes are received, it will follow any geometry link encountered, and subscribe to the base layers of each.

Render renders into a window, which the user is free to resize to a desired size. It renders all objects it knows about into the window, without doing any kind of clipping, culling, visibility testing, or otherwise trying to optimize the workload. Navigation is based on a keyboard and mouse combination, closely mimicking that of so called “first person shooters”; a free-flying model is used where the mouse controls the direction of the camera and four keys on the keyboard allow movement forward/back and left/right.

Render does not do subdivision of geometry, but simply renders the base mesh. It totally ignores material nodes, so there is no lighting, texture mapping, or even proper coloring of polygons.

Conclusions

Render is a very simplistic Verse rendering client, but still manages to be surprisingly useful. It was used to verify the core parts of the data model, has been used in countless demonstrations during the first phase of Verse's development, and even live on stage at the Blender Conference 2004 where Uni-Verse was presented recently.

It is often handy to have a quick rendering client handy, for the cases when one is more interested in the data that is hosted in a server, how it gets there, than looking at it at high quality. Render has been used many times during the development of the Purple scripting/plug-in framework to verify that the correct geometry data is being generated by a graph of plug-ins, for instance.

The brevity of Render's implementation also makes it usable as a potential starting point for developers wishing to learn the basics of Verse geometry storage and rendering, since it is far

easier to read and comprehend through a short program (Render is less than 2,000 lines of C) than a big one. The high-performance rendering client will be based in experience from the simple rendering client.

Availability

Currently, binaries are not maintained for the simple rendering client. Source code is hosted in the public CVS, in the verse-tests module: <http://projects.blender.org/viewcvs/viewcvs.cgi/verse-tests/?cvsroot=verse>.