We introduce jReality, a Java library for creating real-time interactive audiovisual applications with three-dimensional computer graphics and spatialized audio. Applications written for jReality will run unchanged on software and hardware platforms ranging from desktop machines with a single screen and stereo speakers to immersive virtual environments with motion tracking, multiple screens with 3D stereo projection, and multi-channel audio.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—virtual reality, animation; H.5.1[Information Interfaces and Presentation]:Multimedia Information Systems—audio input/output

General Terms: Design

Keywords: virtual reality, immersive environments, interactive spatial audio, Java

1. OVERVIEW

jReality is a library for creating real-time interactive applications with 3D computer graphics and spatialized audio. Applications written for jReality will run unchanged on software and hardware platforms ranging from desktop machines with a single screen and stereo speakers to immersive virtual environments with motion tracking, multiple screens with 3D stereo projection, and arbitrary multi-speaker audio setups. jReality is written in Java and will run on all common operating systems. jReality is metric neutral, supporting hyperbolic and elliptic geometry as well as euclidean geometry.

jReality is open source software, covered by a BSD license. Under constant development since 2003, it offers a robust and reliable codebase supported by an active group of developers. The jReality website (http://www.jreality.de) includes a user forum and a wiki for technical support. The growing developer tutorial currently includes more than seventy-five sample programs illustrating all major features of jReality.

2. DESCRIPTION

jReality is based on a scene graph, a hierarchical representation of a 3D scene. jReality differs from other scene graph libraries in terms of scope and flexibility, achieved with a small but powerful set of building blocks.

One core design feature of jReality is a clear separation of the scene graph (frontend), the rendering components (backends) that translate the scene graph into graphics and sound, and the tool system that handles user interaction. jReality has been designed for thread-safety, so that several backends and tools can operate on one scene graph at the same time.

2.1 Graphics

jReality graphics backends include a pure Java software renderer and a hardware accelerated OpenGL renderer. The former can be deployed remotely where native extensions (required by the OpenGL backend) are not allowed. In cluster-based virtual environments, a distributed backend will display scenes on multiple screens.

Moreover, jReality comes with a number of noninteractive backends, including a backend that exports scenes to PDF as well as a backend that exports scenes to Pixar’s RenderMan for high-quality batch processing. It also includes a U3D backend for embedding interactive 3D content in PDF documents.

jReality reads and writes many popular 3D file formats (OBJ, 3DS, STL, VRML 1.0, Mathematica Graphics 3D, JVX), enabling exchange of data between software packages as well as output to 3D printers.

Graphics rendering is controlled by appearances, a mechanism for defining and inheriting properties such as parameters for point, line, and polygon shaders. jReality shaders can also be customized for individual backends; for example, appearances may specify GLSL shaders for the OpenGL backend that other backends will ignore.

2.2 Audio

jReality supports spatialized audio [1]. Various audio sources (e.g., media players, hardware input, software synthesizers) can be placed in a scene. The audio rendering pipeline of jReality offers auxiliary sends and returns for inserting effects and distance cues like reverberation and distant-dependent attenuation, following [2].
It also implicitly models sound propagation, yielding physically accurate Doppler shifts.

Audio backends will render a stereo signal when running on a desktop system, or 5.1 surround in home theater setups. On a 3D multi-channel speaker rig, jReality will render spatialized audio using Ambisonics [3].

2.4 Plugins

jReality includes a powerful plugin system that allows developers to assemble a user interface from reusable components, including control panels for shader attributes, audio parameters, and a scene graph inspector. These components can be used as a graphical user interface outside the viewer panel as well as inside the scene for use in immersive virtual environments (Figure 1).

3. APPLICATIONS

jReality is the primary software platform of the PORTAL at the Technische Universität Berlin\(^1\) and the VisorLab at the City College of New York.\(^2\) The distribution of jReality comes with tutorial examples as well as a viewer application that lets the user explore objects defined in a number of common 3D file formats.

3.1 Interactive Scientific Simulations

The Demo section of the jReality website lists a large collection of interactive simulations created with jReality. Highlights include a lab for investigating discrete K-surfaces [8], i.e., surfaces of constant negative Gaussian curvature, a lab for investigating the Plateau problem, i.e., the problem of finding minimal surfaces with a given boundary, and a real-time simulation of smoke ring flows using GPGPU techniques [9]. The Virtual Math Labs at TU Berlin\(^3\) contain further examples of scientific visualization with jReality.

Physics simulations are possible with the recent integration of jBullet,\(^4\) the Java port of the Bullet physics engine. This is implemented in a separate open source project and linked from the jReality website. A sample application is available as a Java WebStart.\(^5\)

3.2 Interactive Installations

The Daytar Group\(^6\) has used jReality for creating interactive art, such as the project seidesein. jReality was also featured in the exhibition Imaginary 2008 [4] (Figure 2).\(^7\)

3.3 Animations

jReality was used to create various animations, including a simulation of a Mars flight using data from NASA probes, and the movie The Borromean Rings [6, 5] shown at the opening ceremony of the 2006 International Congress of Mathematicians.

4. INTENDED AUDIENCE

While jReality was originally conceived as a tool for mathematical visualization [7], it has since grown to become a general platform for real-time interactive audio and video. The intended audience includes mathematicians and other scientists with an interest in visualization and sonification, engineers, educators, audiovisual artists, and game developers. With tight integration of 3D graphics and audio as well as support for noneuclidean geometries, jReality opens up creative possibilities that remain largely untapped.

5. ACKNOWLEDGEMENT

The authors wish to acknowledge the contribution of Nils Bleicher, Bernd Gonska, Paul Peters, Holger Pietsch, Markus Schmies, Stefan Sechelmann and Martin Sommer. Stefan Sechelmann\(^8\) has written the plugin package used in jReality.

6. REFERENCES


\(^1\)http://www.tu-berlin.de/3dlabor/ausstattung/
\(^2\)http://math.sci.ccny.cuny.edu/pages?name=VisorLab
\(^3\)http://www.math.tu-berlin.de/geometrie/lab/
\(^4\)http://jbullet.advel.cz
\(^5\)http://www3.math.tu-berlin.de/jreality/webstart/JRbullet.jnlp
\(^6\)http://www.daytar.de/
\(^7\)http://www.imaginary2008.de/
\(^8\)http://www.sechel.de/