

# jReality

## A Java library for real-time interactive 3D graphics and audio

Steffen Weißmann\*  
Technische Universität Berlin  
weissman@math.tu-berlin.de

Charles Gunn  
Technische Universität Berlin  
gunn@math.tu-berlin.de

Peter Brinkmann  
The City College of New York  
brinkman@sci.ccnyc.cuny.edu

Tim Hoffmann  
TU München  
tim.hoffmann@ma.tum.de

Ulrich Pinkall\*  
Technische Universität Berlin  
pinkall@math.tu-berlin.de

### ABSTRACT

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

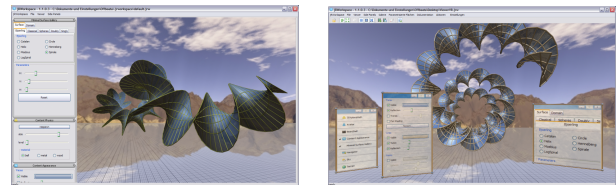
jReality is a library for creating real-time interactive applications with 3D computer graphics and spatialized audio. Applications written for jReality will run unchanged on software and hardware platforms ranging from desktop machines with a single screen and stereo speakers to immersive virtual environments 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.

\*Supported by the DFG Research Center Matheon



**Figure 1: A scene in jReality rendered with the GUI outside the viewer panel (left) and with GUI elements in the scene (right)**

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].

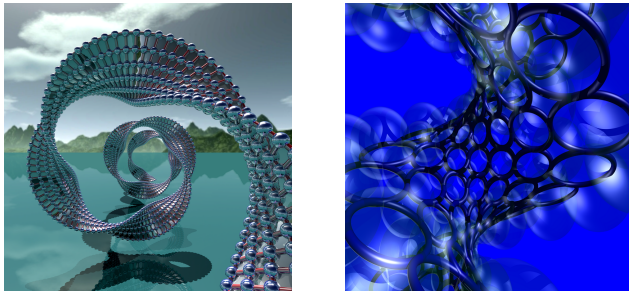


Figure 2: Stills from the jReality exhibit at *Imaginary 2008*.

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].

### 2.3 Tools

The *tool system* of jReality separates the meaning of user interaction from the hardware of the interaction. For instance, a scene may allow the user to rotate an object. The designer of the scene attaches a rotation tool to the object without considering how the user will effect a rotation. At runtime, the tool system determines what input devices are available and matches them with the tools in the scene. On a desktop computer, a rotation will typically be caused by a mouse dragging event. In a virtual environment, motion tracking or wand gestures may trigger a rotation. The application remains the same in both cases.

Currently drivers are available for keyboard, mouse, joystick, Trackd (supporting all popular motion tracking and immersive input devices such as Ascension MotionStar or A.R.T. DTrack systems), Wii Remote, Space Navigator, and GameTrak.

### 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<sup>1</sup> and the VisorLab at the City College of New York.<sup>2</sup> 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

<sup>1</sup><http://www.tu-berlin.de/3dlabor/ausstattung/3d-visualisierung/portal/>

<sup>2</sup><http://math.sci.ccny.cuny.edu/pages?name=VisorLab>

using GPGPU techniques [9]. The *Virtual Math Labs* at TU Berlin<sup>3</sup> contain further examples of scientific visualization with jReality.

Physics simulations are possible with the recent integration of jBullet,<sup>4</sup> 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 Web-Start.<sup>5</sup>

### 3.2 Interactive Installations

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

### 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<sup>8</sup> has written the plugin package used in jReality.

## 6. REFERENCES

- [1] P. Brinkmann and S. Weißmann. Real-time Interactive 3D Audio and Video with jReality. In *Proceedings of the 2009 International Computer Music Conference*. ICMA, 2009.
- [2] R. W. Furse. Spatialisation - Stereo and Ambisonic. In *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming*, 2000.
- [3] M. Gerzon. Surround-sound psychoacoustics. *Wireless World*, 80:486, 1974.
- [4] C. Gunn, T. Hoffmann, N. Schmitt, and U. Pinkall. Differentialgeometrie. In G.-M. Greuel and A. D. Matt, editors, *IMAGINARY*, pages 94–111. MFO, 2008.
- [5] C. Gunn and J. Sullivan. The Borromean Rings: A new logo for the IMU. In K. Polthier, M. Aigner, T. M. Apostol, M. Emmer, C.-H. Hege, and U. Weinberg, editors, “*MathFilm Festival 2008*”, Springer VideoMATH. Springer, 2008.
- [6] C. Gunn and J. Sullivan. The Borromean Rings: A video about the new IMU logo. *Bridges Proceedings (Leeuwarden)*, 2008.
- [7] T. Hoffmann and M. Schmies. *Mathematical Software*, chapter jReality, jtem, and oorange - a way to do math with computers. Springer, 2006.
- [8] U. Pinkall. Designing cylinders with constant negative curvature. In A. Bobenko, P. Schröder, J. Sullivan, and G. Ziegler, editors, *Discrete Differential Geometry*, pages 57–67. Birkhäuser, 2008.
- [9] U. Pinkall, B. Springborn, and S. Weißmann. A new doubly discrete analogue of smoke ring flow and the real time simulation of fluid flow. *Journal of Physics A: Mathematical and Theoretical*, 40(42):12563–12576, 2007.

<sup>3</sup><http://www.math.tu-berlin.de/geometrie/lab/>

<sup>4</sup><http://jbullet.advel.cz>

<sup>5</sup><http://www3.math.tu-berlin.de/jreality/webstart/JRBullet.jnlp>

<sup>6</sup><http://www.daytar.de/>

<sup>7</sup><http://www.imaginary2008.de/>

<sup>8</sup><http://www.sechel.de/>