17

3D Graphics Kit

Library:	libMedia_s.a
Header File Directory:	/NextDeveloper/Headers/3Dkit
Import:	3Dkit/3Dkit.h

Introduction

The 3D Graphics Kit enables NEXTSTEP applications to model and render 3-dimensional scenes. Much as the Application Kit's 2D graphics capabilities are based on the Display PostScript interpreter, the 3D Kit's capabilities are based on the Interactive RenderMan renderer.   There are both similarities and differences in the inner workings of the two implementations.

One similarity is that both are implemented with a client-server model, in which client applications send drawing code to the Window Server, which does the actual drawing.   Another similarity is that N3DCameraÐthe 3D Kit's ViewÐgenerates *all* drawing code, both 2D and 3D, when its **drawSelf:** method is invoked.   This keeps the Application Kit's display mechanism intact for both PostScript and RenderMan drawing.

One difference in the implementations is in the code generated for drawing.   For 2D drawing, a View sends PostScript code to the Window Server's Display PostScript interpreter.   For 3D drawing, a View sends RenderMan Interface Bytestream (RIB) code to the Window Server's Interactive RenderMan renderer.

The PostScript language is frequently referred to as a *page description language*; The RenderMan language can be thought of as a *scene description language*.   It provides graphics primitives, lighting specification, camera controls, and other features required for 3D scene description.   This documentation assumes you are familiar with the RenderMan language; for an introduction to the language, see *The RenderMan Companion* by Steve Upstill, published by Addison-Wesley.

The RenderMan Interface and 3D Renderers

The RenderMan Interface is a standard API for 3D scene description.   One of the main features of the RenderMan Interface is that it separates *modeling* from *rendering*.   A modeling program stores data for the objects in a 3D scene and generates RIB code to describe that scene to a renderer.   The level of detail in the model is fixed in the data stored by the modeler.   The quality of rendering is determined by the renderer selected and the rendering techniques selected for that renderer.

The 3D Kit uses two separate renderers: the interactive renderer for display and the photorealistic renderer for printed output.

The Interactive Renderer

To draw 3D scenes on-screen, a 3D Kit application sends its RIB output to the Interactive RenderMan renderer. For optimal drawing in response to user actions, the interactive renderer doesn't implement some features of the full RenderMan language. However, it does process all RIB code without error, ignoring attributes and options that it doesn't implement. As one example, shaders written in the RenderMan Shading Language aren't applied to surfaces by the interactive renderer (except for a limited group of standard shaders).

So that multiple applications can render 3D scenes simultaneously, the interactive renderer implements additions to the RenderMan language for creating, selecting, and destroying contexts. Client applications create handles for their rendering contexts, select the appropriate context before they begin generating drawing code, and destroy contexts when they are finished with them. For the most part, interactive rendering contexts are managed by the 3D Kit, so you rarely have to deal with them in your code.

A specification for the interactive renderer, including descriptions of new RenderMan procedures it implements and standard RenderMan procedures it ignores, can be found in the release note **/NextLibrary/Documentation/NextDev/Pixar/QRMSpec.rtfd**.

The Photorealistic Renderer

The Application Kit's printing mechanism is extended by the 3D Kit to enable RIB output to be correctly incorporated into a print stream. When rendering a 3D image to be printed on a page or saved in a file, 3D Kit applications send their RIB output to the PhotoRealistic RenderMan renderer. The photorealistic renderer generates TIFF image data, which is then incorporated into the PostScript print stream.

The photorealistic renderer supports the full RenderMan standard (with a few minor exceptions), so the images it generates display the detail and features specified in the original model. The photorealistic renderer operates as a separate process. It starts when invoked by a 3D Kit client, and stops when the image based on that RIB has been rendered. For speedier rendering, the 3D Kit supports photorealistic rendering in multiple processes on multiple hosts.