TP001: A Scene Data Management Approach to Real Time 3D Software Applications

Introduction

The concept of a scene management approach to 3D applications is not completely new. RealiMation has been shipping as a commercial product now for just over two years. This document gives a generic description of what is meant by “scene management”.

What is Scene Data Management?

It starts with a requirement. This may be anything from a very specific scientific application to visualize data in 3D, right through to a full blown application to be sold commercially in very high volume. What ever the requirement is, we have assumed that there is a need for real time 3D computer graphics within it.

Scene Data Management in this context refers to the concept of a software application operating at a higher level than they would need to ordinarily. I.e. dealing with high level objects within a ‘scene’ as opposed to getting down deep and dirty and dealing with painting pixels.

Most people writing 3D applications are used to using a software/hardware rendering engine such as OpenGL or Direct-3D. These technologies deal with display devices that paint pixels on the screen and themselves have a higher level input in the form of triangles or polygons. A scene management engine takes the programmer to a higher level, being able to deal with a high level construct such as an ‘attacker’ a ‘tank’ or ‘enemy no 346’. In this situation, attacker may well be an attack helicopter in the shape of an Apache or a Hind. It will have rotor blades that are moving around at a pre-determined rate and when it is moved, the programmer can focus on moving ***it*** - i.e. the ‘attacker’ object rather than dealing with the triangles and subsequent pixels that make it up. Moreover, a comprehensive scene management API will also have facilities to deal with the rotation of the blades - invisibly to the programmer if required.

The scene management tool enables the developer to concentrate on the application *content*, without having to worry about how that content is delivered to the end user.

Things to look for in a Scene Data Management toolset:

Capability. With good design and implementation?

Thoroughness. In design depth. Can it deal with those awkward situations that only I need?

Extensibility. Is it easy to add my own functionality if required?

Development Process. Is the methodology clearly defined?

Interfacing Capabilities. Does the toolkit interface nicely with my design and modeling tools?

Real Time Preparation aides. Does the toolkit provide additional tools to help me prepare the models and data for real time delivery?

Cross Platform support. What is the development environment based on? What platforms can be delivered on?

COST EFFECTIVENESS - How much would it cost me to write my own toolset? Do you *really* want to write one?