SceneGraph XML Overview

Table of Contents

The term *SceneGraph* refers to a design algorithm and associated programming constructs that are widely used in computer graphics systems, such as video games. A SceneGraph uses a tree structure of image element *nodes* to define an interactive scene that is traversed to render an image, with the position of each node in the tree determining the z-axis rendering of the node image element; nodes lower in the tree structure are rendered over nodes higher in the tree. Each node in the tree structure is an object whose state is stored as attributes in a set of *fields*. As the tree is traversed, the rendering state is accumulated, so that the state of a parent node can control how the child nodes are rendered.

Grouping and Inheritance

This node tree structure allows the nodes in the tree to be logically grouped and manipulated. For example, a node in the tree may have an opacity field (*opacity* being the inverse of transparency). When set to a value of 0.5, that node and all of its descendents can have their opacity multiplied by 0.5.

XML Definition of a SceneGraph Tree

In general, the attribute fields of a node contain the data that is required to create the appearance and behavior of the node. Roku SceneGraph uses XML files to define the attributes of all SceneGraph nodes. The attribute fields are strongly typed and can include arbitrary data, including floating point numbers, integers, strings, Boolean values, 2D vectors, and pointers to other nodes. Fields can have animations attached to them that allow the property represented by the field to be smoothly interpolated through a set of key values.

Renderable/Non-Renderable Nodes

In Roku SceneGraph, nodes can either be *renderable* or not. Renderable nodes are arranged in the tree structure that is traversed during rendering to draw the scene. Non-renderable nodes are ignored during the render traversal, and are used to describe other information used by the scene, such as timers, animations, and data to be displayed. In general, the term node will be used interchangably for both renderable and non-renderable nodes unless it is important to distinguish between the two.

In the Roku SceneGraph implementation, there are only three basic types of nodes that actually draw something.

- A **Rectangle** node renders a rectangle to the display screen
- A **Label** node renders a formatted string of text to the display screen
- A **Poster** node renders a graphical image to the display screen

The power of the Roku SceneGraph architecture is that the three basic renderable node classes, and the supporting non-renderable node classes, can be arbitrarily composed into new node classes that encapsulate more complex appearance and behavior, including common user interface widgets such as buttons, lists, grids, virtual keyboards, and dialogs.

Key Events/Rendering Optimization

In addition to adding logical structure to graphical user interface elements and supporting rendering/animation, the SceneGraph algorithm is useful in other ways including:

- **Supporting key event propagation**
  A node in the graph is given remote control key focus and receives key events. If that node does not handle the particular key event, the event is passed up the SceneGraph to its parent node, until it reaches a node that handles the event.

- **Rendering optimization**
  Roku SceneGraph uses a render culling algorithm that allows entire branches of the SceneGraph to not be traversed for rendering if the bounding rectangle of the root of that branch is outside the viewable area of the TV display screen, allowing much quicker and more efficient scene rendering.

- SceneGraph Coordinate Systems
- Other Inherited Properties
- Node Field Observers
- Remote Control Events
