ivJava

Overview:

This project aims to demonstrate usability of Inventor from Java. Java awt provides a canvas component that allows the user to customize rendering on the canvas by exposing a native interface callable using JNI. In this project a canvas called ivCanvas is coded using JNI. This canvas inherits from java awt Canvas and so can be used as a component in any java user interface programs. ivCanvas' paint method has a native implementation, enabling it to render the canvas using OpenGL. This paint method makes it possible to render an inventor scene on the canvas. The canvas can load any scene graph in iv file format, and render the corresponding scene. Thus the viewer functionality of inventor is available in java by using JNI calls to bind to the C++ based inventor methods.

To show how the idea works, ivCanvas class is implemented which extends Canvas class (refer ivCanvas.java). Any event inside ivCanvas is captured by respective action listeners defined in ivCanvas class. The event is then serialized to a string and passed to C++ for conversion to inventor based event objects, which are then passed over to inventor. Thus any graphs containing draggers and other interactive elements will receive all events on the canvas in inventor format.

As an example MainFrame class is created which extends Frame class. The form has a file menu with two options "Open" and "Demo". Demo is used to load a demo graph and Open option is to load .iv file. Demo graph demonstrates handling of both mouse and keyboard events. It consists of a cone with a dragger. While the cone moves using the arrow keys, the mouse movements are handled by the dragger and reflected in the cone's radius.

Prerequisite:

The following Installations are required

1. Java 1.6 or later version.
2. SGI open inventor.

Description:

Class	Description	Function	
		Name	Description
MainFrame	Example class to test the functionality of myWindow class.	initComponents, actionPerformed to initialize the components of form and to handle events within Canvas	
RunIvJava	Defines main function and creates the MainFrame class object.	Main class to instantiate and invoke MainFrame.	

ivCanvas	It extends Canvas class. The object of this class is a window which handles all the events within this window. After handling the event, paint method is called to repaint the Window.	cleanupOpenGL	Native method call to Release the device context for current window.
		formKeyPressed	Any keypressed event inside the window calls formKeyPressed which pass the event to native method keyPressedEvent for processing
		formKeyReleased	Any keyReleased event inside the window calls formKeyReleased which passes the event to native method keyReleasedEvent for processing
		formMouseEvent	Any Mouse event inside the window calls formMouseEvent which passes the event to native method mouseEventHandler for processing
		Initialize	Call initialSetup for initial setting of window and also add Event listeners to the window
		get	To get the value of property passed as parameter.
		initialSetup	Native method to do all the initial setting of window
		initializeOpenGL	Use the Device Context of current window to set the default pixel format, window size, viewport region, background color, etc and activate the SceneManager for this window
		keyPressedEvent/ keyReleasedEvent	Called when any Key is Pressed. Create the SoKeyboard event object from java passed event description and process the event.
		mouseEventHandler	Called when any mouse action take place in window. Creates the SoMouseButtonEvent or SoLocation2Event based on the event description passed to it.
		Paint	Override the paint method of canvas class to add the user defined functionality of initializeOpenGl and paintOpenGL.
		paintOpenGL	Native Java method which render the graph. Call render function to render the graph using SoSceneManager object for current window.
		loadGraphOpenGL	Native Java code to load the given graph by calling setSceneGraph from sceneManager.
		set	To set the value of property passed as parameter
		setBackgroundColor	To set the background color of the window
		setViewportRegion	To change the Viewport of the sceneManager