Three-Dimensional Printing on a Budget:
A Classroom-Friendly Technique for
Viewing Solid Objects

Developer’s Reference Manual

Diana Butter  Jeremy Garcia  Ryan Lewis
Tyler Nielsen

Sponsored by:
Michael Eisenberg
Department of Computer Science
University of Colorado, Boulder
Contents

1 SPECTRE  
2 SPECTRE Namespace Index  
3 SPECTRE Hierarchical Index  
4 SPECTRE Compound Index  
5 SPECTRE File Index  
6 SPECTRE Namespace Documentation  
7 SPECTRE Class Documentation  
8 SPECTRE File Documentation
1 SPECTRE

1.1 Introduction

SPECTRE is a software system for “affordable” 3D printing. It consists of a simple 3D CAD interface for creating scenes composed of 3D objects, and a utility to “slice” the created scene to generate a set of transparencies that can be stacked to view the scene in three dimensions. Please consult the User Reference Manual for additional information.

This is the developer documentation for SPECTRE. It is divided into two main sections: the first section provides a high-level overview of the SPECTRE source code, including how to install and compile it. In addition, it discusses the hardware and software requirements for building and running SPECTRE. The second section provides a detailed description of all the files, classes, methods, and member variables in the SPECTRE source code. This section is generated directly from the source by Doxygen.

1.2 SPECTRE Installation

The SPECTRE source code is distributed as a compressed tar file. You can obtain the source code from the SPECTRE website: http://www.mithander.com/spectre. Once you have obtained the source code, use the following procedure to install and compile it:

1. Determine if you have the proper hardware and software environment
2. Expand the SPECTRE source code
3. Compile the source code

Each of these steps is described in the proceeding sections.

1.2.1 System Requirements

Although SPECTRE’s target platform was Microsoft Windows 98, it has been successfully compiled and run on GNU/Linux. It should be possible to compile SPECTRE on any platform with all the following:

- Java SDK 1.4 or higher
- Java3D SDK 1.2 or higher
- JavaCC 2.1
- GNU Make

The following utilities are required to compile the SPECTRE documentation:
1.3 Overview of the SPECTRE Source

- \LaTeX, BibTeX, and dvips
- Doxygen 1.2
- fig2dev 3.2

The runtime software environment must include the following:

- Java JRE 1.4 or higher
- Java3D 1.2 Runtime or higher
- A color inkjet printer

1.2.2 Expanding the SPECTRE Source

As stated above, the SPECTRE source code is distributed as a compressed tar archive. To expand the source files, use this command in the directory where the archive resides: \texttt{tar xzvf filename.tar.gz}. This will create a new directory called “spectre” which contains the source files needed to generate the executable and documentation.

1.2.3 Compiling the SPECTRE Source

Ensure that \texttt{javac}, \texttt{javacc}, \texttt{make}, \texttt{doxygen}, \texttt{latex}, \texttt{dvips}, \texttt{bibtex}, \texttt{fig2dev}, \texttt{latex2html}, \texttt{dvips}, and \texttt{dvipdf} are in your current path. In addition, ensure that the Java3D SDK has been properly installed. Then, change to the top-level SPECTRE directory: \texttt{cd spectre}. Finally, run \texttt{make all} to build the SPECTRE executable and documentation.

Using the above procedure generates the following files:

- html/spectre.jar
- Several documents in printable and browsable format under html/doc

Load the file html/index.html in a browser to view the generated documents. Open the jar file using your Java interpreter to start SPECTRE.

1.3 Overview of the SPECTRE Source

This section provides a discussion of the organization of the source code and mentions the known problems with SPECTRE.

1.3.1 Source Code Organization

The SPECTRE source code resides entirely in the \texttt{src} directory. This directory contains five subdirectories; each subdirectory corresponds to a Java package,
with the exception of the images directory. The list below provides a brief description of each directory. For more information, consult each package’s detailed documentation later in this manual.

- disp source for managing and displaying the current scene
- images images for the toolbars and startup splash screen
- io source for reading saved files from disk
- obj source for creating and manipulating objects
- win source for creating windows and handling user events

The SPECTRE documentation resides in two places. The source documentation (which you are currently reading), is contained in the src directory. This documentation is generated using Doxygen from comments in the actual source code. All other documents are written in LaTeX, and reside in subdirectories of the doc directory. These documents are:

- arch An (outdated) description of the SPECTRE system architecture
- req An (outdated) description of the SPECTRE initial requirements
- design A description of the SPECTRE architecture and design
- reqspec A listing of the SPECTRE software requirements
- testplan The SPECTRE test plan
- tutorial The User Tutorial
- reference The User Reference Manual
- device The Device Construction Tutorial

In addition, the directory pres contains slides from various presentations, and the directory images contains pictures and figures used in the documentation and slides.

1.3.2 Problems

- When loading saved scenes from a file, SPECTRE does not save memory by reusing identical poly meshes in different SceneGraphNodes. This would require a change in the SPECTRE file format as well as a change in the parser.
- When saving a scene to file, a Scene’s toString() method is called. For large scenes, this method can use excessive amounts of memory. It would be better to pass the Scene an output stream, and have it write each SceneGraphNode to the output stream.
- Although there is support in the code and file format for recursive SceneGraphs (which would allow grouping of objects), this is currently not supported in the user interface.
2  SPECTRE Namespace Index

2.1  SPECTRE Package List

Here are the packages with brief descriptions (if available):

- `com.sun.j3d.util.behaviors.mouse` 14
- `com.sun.j3d.util.geometry` 14
- `com.sun.j3d.util.universe` 14
- `disp` (Provides the needed implementation to store/retrieve a scene) 14
- `io` (Handle all the disk io) 14
- `java.awt` 16
- `java.awt.event` 16
- `java.awt.print` 16
- `java.io` 16
- `java.util.zip` 16
- `javax.media.j3d` 16
- `javax.print.attribute` 16
- `javax.print.attribute.standard` 16
- `javax.swing` 16
- `javax.vecmath` 16
- `obj` (Handles simple object creation and manipulation) 16
- `win` (Provides for all user interface related issues) 17

3  SPECTRE Hierarchical Index

3.1  SPECTRE Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>win.ControlWindow</td>
<td>18</td>
</tr>
<tr>
<td>win.Display</td>
<td>31</td>
</tr>
<tr>
<td>disp.Geometry</td>
<td>36</td>
</tr>
<tr>
<td>win.J3DWindow</td>
<td>43</td>
</tr>
<tr>
<td>win.MenuHandler</td>
<td>47</td>
</tr>
<tr>
<td>win.ObjDialog</td>
<td>52</td>
</tr>
<tr>
<td>obj.ObjFactory</td>
<td>67</td>
</tr>
<tr>
<td>win.OptionsDialog</td>
<td>70</td>
</tr>
<tr>
<td>io.ParseException</td>
<td>72</td>
</tr>
<tr>
<td>io.ParserConstants</td>
<td>85</td>
</tr>
<tr>
<td>io.Parser</td>
<td>76</td>
</tr>
<tr>
<td>io.ParserTokenManager</td>
<td>89</td>
</tr>
<tr>
<td>io.ParserInterface</td>
<td>88</td>
</tr>
<tr>
<td>disp.Scene</td>
<td>127</td>
</tr>
<tr>
<td>disp.PickablePolygon</td>
<td>102</td>
</tr>
<tr>
<td>obj.Polymesh</td>
<td>107</td>
</tr>
<tr>
<td>obj.Polymesh,Line</td>
<td>117</td>
</tr>
<tr>
<td>obj.Polymesh,Plane</td>
<td>119</td>
</tr>
<tr>
<td>obj.Polymesh,Tri</td>
<td>121</td>
</tr>
<tr>
<td>obj.Polymesh,Vertex</td>
<td>122</td>
</tr>
<tr>
<td>disp.Render</td>
<td>125</td>
</tr>
<tr>
<td>obj.SceneGraph</td>
<td>140</td>
</tr>
<tr>
<td>obj.SceneGraphNode</td>
<td>145</td>
</tr>
<tr>
<td>io.SimpleCharStream</td>
<td>154</td>
</tr>
<tr>
<td>io.SpectreFileError</td>
<td>165</td>
</tr>
<tr>
<td>Class</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>win.SplashWindow</td>
<td>166</td>
</tr>
<tr>
<td>io.Token</td>
<td>172</td>
</tr>
<tr>
<td>io.TokenMgrError</td>
<td>176</td>
</tr>
<tr>
<td>obj.Transform</td>
<td>180</td>
</tr>
<tr>
<td>obj.TransformFactory</td>
<td>183</td>
</tr>
<tr>
<td>ActionEvent</td>
<td>??</td>
</tr>
<tr>
<td>ActionListener</td>
<td>??</td>
</tr>
<tr>
<td>Applet</td>
<td>??</td>
</tr>
<tr>
<td>Arrays</td>
<td>??</td>
</tr>
<tr>
<td>BasicStroke</td>
<td>??</td>
</tr>
<tr>
<td>BevelBorder</td>
<td>??</td>
</tr>
<tr>
<td>BorderFactory</td>
<td>??</td>
</tr>
<tr>
<td>BorderLayout</td>
<td>??</td>
</tr>
<tr>
<td>BorderLayout</td>
<td>??</td>
</tr>
<tr>
<td>BorderLayout</td>
<td>??</td>
</tr>
<tr>
<td>Box</td>
<td>??</td>
</tr>
<tr>
<td>BoxLayout</td>
<td>??</td>
</tr>
<tr>
<td>ButtonGroup</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
<tr>
<td>Color</td>
<td>??</td>
</tr>
</tbody>
</table>

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
### 3.1 SPECTRE Class Hierarchy

<table>
<thead>
<tr>
<th>Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparator</td>
<td>??</td>
</tr>
<tr>
<td>obj, Polymesh, HullSort</td>
<td>114</td>
</tr>
<tr>
<td>Component</td>
<td>??</td>
</tr>
<tr>
<td>Container</td>
<td>??</td>
</tr>
<tr>
<td>Container</td>
<td>??</td>
</tr>
<tr>
<td>Dimension</td>
<td>??</td>
</tr>
<tr>
<td>Dimension</td>
<td>??</td>
</tr>
<tr>
<td>Dimension</td>
<td>??</td>
</tr>
<tr>
<td>Dimension</td>
<td>??</td>
</tr>
<tr>
<td>Double</td>
<td>??</td>
</tr>
<tr>
<td>File</td>
<td>??</td>
</tr>
<tr>
<td>File</td>
<td>??</td>
</tr>
<tr>
<td>File</td>
<td>??</td>
</tr>
<tr>
<td>File</td>
<td>??</td>
</tr>
<tr>
<td>FileFilter</td>
<td>??</td>
</tr>
<tr>
<td>win, ExtensionFilter</td>
<td>34</td>
</tr>
<tr>
<td>FileInputStream</td>
<td>??</td>
</tr>
<tr>
<td>FileInputStream</td>
<td>??</td>
</tr>
<tr>
<td>Graphics</td>
<td>??</td>
</tr>
<tr>
<td>Graphics2D</td>
<td>??</td>
</tr>
<tr>
<td>Graphics2D</td>
<td>??</td>
</tr>
<tr>
<td>GraphicsConfiguration</td>
<td>??</td>
</tr>
<tr>
<td>Hashtable</td>
<td>??</td>
</tr>
<tr>
<td>Hashtable</td>
<td>??</td>
</tr>
<tr>
<td>InputEvent</td>
<td>??</td>
</tr>
<tr>
<td>Iterator</td>
<td>??</td>
</tr>
<tr>
<td>Iterator</td>
<td>??</td>
</tr>
</tbody>
</table>

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
<table>
<thead>
<tr>
<th>Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterator</td>
<td>??</td>
</tr>
<tr>
<td>Iterator</td>
<td>??</td>
</tr>
<tr>
<td>Iterator</td>
<td>??</td>
</tr>
<tr>
<td>JApplet</td>
<td>??</td>
</tr>
<tr>
<td>win,MainWindow</td>
<td>45</td>
</tr>
<tr>
<td>JButton</td>
<td>??</td>
</tr>
<tr>
<td>JColorChooser</td>
<td>??</td>
</tr>
<tr>
<td>JDialog</td>
<td>??</td>
</tr>
<tr>
<td>JDialog</td>
<td>??</td>
</tr>
<tr>
<td>JLabel</td>
<td>??</td>
</tr>
<tr>
<td>win,StatusBar,StatusPane</td>
<td>171</td>
</tr>
<tr>
<td>JOptionPane</td>
<td>??</td>
</tr>
<tr>
<td>JPanel</td>
<td>??</td>
</tr>
<tr>
<td>win,StatusBar</td>
<td>166</td>
</tr>
<tr>
<td>win,StatusBar,ColorPane</td>
<td>169</td>
</tr>
<tr>
<td>JRadioButton</td>
<td>??</td>
</tr>
<tr>
<td>JTextComponent</td>
<td>??</td>
</tr>
<tr>
<td>JTextField</td>
<td>??</td>
</tr>
<tr>
<td>Line2D</td>
<td>??</td>
</tr>
<tr>
<td>LinkedList</td>
<td>??</td>
</tr>
<tr>
<td>LinkedList</td>
<td>??</td>
</tr>
<tr>
<td>LinkedList</td>
<td>??</td>
</tr>
<tr>
<td>LinkedList</td>
<td>??</td>
</tr>
<tr>
<td>LinkedList</td>
<td>??</td>
</tr>
<tr>
<td>LinkedList</td>
<td>??</td>
</tr>
<tr>
<td>LinkedList</td>
<td>??</td>
</tr>
<tr>
<td>MainFrame</td>
<td>??</td>
</tr>
<tr>
<td>Class</td>
<td>Reference</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Matrix4d</td>
<td>??</td>
</tr>
<tr>
<td>Matrix4d</td>
<td>??</td>
</tr>
<tr>
<td>MouseAdapter</td>
<td>??</td>
</tr>
<tr>
<td>win.PickingMouseListener</td>
<td>105</td>
</tr>
<tr>
<td>MouseEvent</td>
<td>??</td>
</tr>
<tr>
<td>Point</td>
<td>??</td>
</tr>
<tr>
<td>Point</td>
<td>??</td>
</tr>
<tr>
<td>Polygon</td>
<td>??</td>
</tr>
<tr>
<td>Polygon</td>
<td>??</td>
</tr>
<tr>
<td>PrintService</td>
<td>??</td>
</tr>
<tr>
<td>Rectangle</td>
<td>??</td>
</tr>
<tr>
<td>Rectangle</td>
<td>??</td>
</tr>
<tr>
<td>RenderingHints</td>
<td>??</td>
</tr>
<tr>
<td>URL</td>
<td>??</td>
</tr>
<tr>
<td>URL</td>
<td>??</td>
</tr>
<tr>
<td>URL</td>
<td>??</td>
</tr>
<tr>
<td>Vector</td>
<td>??</td>
</tr>
<tr>
<td>Vector</td>
<td>??</td>
</tr>
<tr>
<td>Vector</td>
<td>??</td>
</tr>
<tr>
<td>Vector</td>
<td>??</td>
</tr>
<tr>
<td>Vector</td>
<td>??</td>
</tr>
<tr>
<td>Vector</td>
<td>??</td>
</tr>
<tr>
<td>Window</td>
<td>??</td>
</tr>
<tr>
<td>ZipEntry</td>
<td>??</td>
</tr>
<tr>
<td>ZipInputStream</td>
<td>??</td>
</tr>
</tbody>
</table>
4 SPECTRE Compound Index

4.1 SPECTRE Compound List

Here are the classes, structs, unions and interfaces with brief descriptions:

- `win.ControlWindow` (Main view a user will interact with) 18
- `win.Display` (Manages 2D and 3D windows in UI) 31
- `win.ExtensionFilter` (Handles extension options in a save/open dialog box) 34
- `disp.Geometry` (Static class to handle conversion from inches to pixels) 36
- `win.J3DWindow` (3D view of a scene) 43
- `win.MainWindow` (This does nothing other than open up the control window) 45
- `win.MenuHandler` (Handles menu accesses) 47
- `win.ObjDialog` (Creates a multi-use dialog box) 52
- `obj.ObjFactory` (Creates Sphere, Cube, Cone, and Cylinders from Polymeshes) 67
- `win.OptionsDialog` (Class to display the options dialog) 70
- `io.ParseException` (This exception is thrown when parse errors are encountered) 72
- `io.Parser` (Class to parse spectre files, and add the parsed objects to a scene) 76
- `io.ParserConstants` 85
- `io.ParserInterface` (Interface that allows the parser to add objects to the current scene) 88
- `io.ParserTokenManager` 89
- `disp.PickablePolygon` (Creates a drawable polygon with a reference back to the SceneGraph) 102
- `win.PickingMouseListener` (Handles events thrown by user input via mouse) 105
obj.Polymesh (A mesh of triangles enclosing a solid region of space) 107

obj.Polymesh.HullSort (A comparator that can sort points in counter-clockwise order around a point) 114

obj.Polymesh.Line (A line segment in 3D space) 117

obj.Polymesh.Plane (A plane in 3D space) 119

obj.Polymesh.Tri (A triangle in 3D space) 121

obj.Polymesh.Vertex (A vertex in 3D space using homogeneous coordinates) 122

disp.Render (Holds data needed to render a view) 125

disp.Scene (This holds all the information about a scene) 127

obj.SceneGraphNode (Nodes that compose a SceneGraph (p. 140)) 145

io.SimpleCharStream (An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (without unicode processing)) 154

io.SpectreFileError (Exception used by the parser to indicate file errors encountered while parsing a file) 165

win.SplashWindow 166

win.StatusBar (Class for status bar; denotes slice viewed) 166

win.StatusBar.ColorPane 169

win.StatusBar.StatusPane (Panel for display in status bar) 171

io.Token (Describes the input token stream) 172

io.TokenMgrError 176

obj.Transform (Wrapper around a 4x4 transformation matrix for 3D homogeneous coordinates) 180

obj.TransformFactory (Contains several static methods that return common Transformations) 183
5 SPECTRE File Index

5.1 SPECTRE File List

Here is a list of all files with brief descriptions:

ControlWindow.java (The main SPECTRE window, the slicer, handles user input and displays slices) 186

disp.dox 187

Display.java (Handles communication between 2D and 3D windows, and provides a wrapper around the 3D client, so Java3D is not a system requirement) 187

ExtensionFilter.java (Handles extension options in save/open dialog box) 188

Geometry.java (Implements a class consisting of static functions useful for drawing) 188

io.dox 189

J3DWindow.java (Creates a window for viewing scenes using Java3D) 189

mainpage.dox 189

MainWindow.java (Opens the ControlWindow, whether opened as an applet or by a call to main) 189

MenuHandler.java (Handles majority of menu events) 190

obj.dox 191

ObjDialog.java (Creates majority of dialog boxes, for on-the-fly object creation as well as for object manipulations) 191

ObjFactory.java (Routines to generate Polymeshes for common objects) 191

OptionsDialog.java (Dialog box for setting Scene-wide options) 192

ParseException.java 193

Parser.java 193

ParserConstants.java 193
ParserInterface.java (Defines the ParserInterface used to add objects to the current scene) 193

ParserTokenManager.java 194

PickablePolygon.java (A drawable polygon coupled with a reference back to the SceneGraph) 194

PickingMouseListener.java (Used to 'pick' a polygon from the current scene) 194

Polymesh.java (Implements a vertex-edge-face representation of 3D objects) 195

Render.java (Class to hold data needed to render a view) 195

Scene.java (Holds data relevant to the global scene, rather than individual windows) 196

SceneGraph.java (Implement a scene graph composed of Poly-

mesh objects with Transformations applied to them) 197

SceneGraphNode.java (Implements the SceneGraphNode class,

which makes up a SceneGraph) 197

SimpleCharStream.java 198

SpectreFileError.java (Defines a class of exceptions the parser can use to indicate various errors encountered when parsing a file) 198

SplashWindow.java (This class displayes a window while spectre is loading) 199

StatusBar.java (Creates a status bar for ControlWindow which denotes current slice viewed) 199

Token.java 200

TokenMgrError.java 200

Transform.java (Implements a Transformation that can be applied to SceneGraphNodes) 200

TransformFactory.java (Create Transforms for common pur-

poses) 201

win.dox 202
6 SPECTRE Namespace Documentation

6.1 Package com.sun.j3d.utils.behaviors.mouse

6.2 Package com.sun.j3d.utils.geometry

6.3 Package com.sun.j3d.utils.universe

6.4 Package disp

Provides the needed implementation to store/retrieve a scene.

Compounds

- class Geometry
  
  Static class to handle conversion from inches to pixels.

- class PickablePolygon
  
  Creates a drawable polygon with a reference back to the SceneGraph.

- class Render
  
  Holds data needed to render a view.

- class Scene

  This holds all the information about a scene.

6.4.1 Detailed Description

Provides the needed implementation to store/retrieve a scene.

Handles translations, filename, render options. Also responsible for storing all objects in the scene and knowing how to access them.

6.5 Package io

Handle all the disk io.

Compounds

- class ParseException

  This exception is thrown when parse errors are encountered.
• class Parser
  
  Class to parse spectre files, and add the parsed objects to a scene.

• interface ParserConstants
  • interface ParserInterface

  Interface that allows the parser to add objects to the current scene.

• class ParserTokenManager
  • class SimpleCharStream

  An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (without unicode processing).

• class SpectreFileError

  Exception used by the parser to indicate file errors encountered while parsing a file.

• class Token

  Describes the input token stream.

• class TokenMgrError

6.5.1 Detailed Description

Handle all the disk io.
6.6 Package java.awt
6.7 Package java.awt.event
6.8 Package java.awt.print
6.9 Package java.io
6.10 Package java.util.zip
6.11 Package javax.media.j3d
6.12 Package javax.print.attribute
6.13 Package javax.print.attribute.standard
6.14 Package javax.swing
6.15 Package javax.vecmath
6.16 Package obj

Handles simple object creation and manipulation.

Compounds

- class **HullSort**
  
  *A comparator that can sort points in counter-clockwise order around a point.*

- class **Plane**
  
  *A plane in 3D space.*

- class **Line**
  
  *A line segment in 3D space.*

- class **ObjFactory**
  
  *Creates Sphere, Cube, Cone, and Cylinders from Polymeshes.*

- class **Polymesh**
  
  *A mesh of triangles enclosing a solid region of space.*

- class **Vertex**
6.17 Package win

A vertex in 3D space using homogeneous coordinates.

- class Tri
  A triangle in 3D space.

- class SceneGraph
  Controls access to all the SceneGraphNodes.

- class SceneGraphNode
  Nodes that compose a SceneGraph (p.140).

- class Transform
  Wrapper around a 4x4 transformation matrix for 3D homogeneous coordinates.

- class TransformFactory
  Contains several static methods that return common Transformations.

6.16.1 Detailed Description

Handles simple object creation and manipulation.

6.17 Package win

Provides for all user interface related issues.

Compounds

- class ColorPane
- class ControlWindow
  Main view a user will interact with

- class Display
  Manages 2D and 3D windows in UI.

- class ExtensionFilter
  Handles extension options in a save/open dialog box.

- class J3DWindow
  3D view of a scene.
• class MainWindow
  
  This does nothing other than open up the control window.

• class MenuHandler
  
  Handles menu accesses.

• class ObjDialog
  
  Creates a multi-use dialog box.

• class OptionsDialog
  
  Class to display the options dialog.

• class PickingMouseListener
  
  Handles events thrown by user input via mouse.

• class SplashWindow
  
  Class for status bar; denotes slice viewed.

• class StatusPane

  Panel for display in status bar.

6.17.1 Detailed Description

Provides for all user interface related issues.

Handles menu selection, window drawing and toolbar creation.

7 SPECTRE Class Documentation

7.1 win.ControlWindow Class Reference

Main view a user will interact with.

Collaboration diagram for win.ControlWindow:
Public Methods

- void **showWindow** ()
  
  *Setup the window, then display it.*

- void **hide** ()
  
  *Hide window when needed.*

- void **updateWindow** ()
  
  *Redraw window when needed.*

- int **print** (Graphics g, PageFormat pageFormat, int pageIndex)
  
  *Print on demand.*

- int **getNumberOfPages** ()
7.1 win.ControlWindow Class Reference

Returns the number of pages to print.

- PageFormat `getPageFormat` (int pageIndex)
  Gets the format for a page.

- Printable `getPrintable` (int pageIndex)
  Gets the object responsible for printing a page.

Static Public Methods

- void `drawFrame` (Graphics g, Rectangle r, Render rv, Vector objRef)
  Render a frame in a window.

Public Attributes

- Render view
  Section of ControlWindow (p.18) where slices are drawn.

- boolean `dirty` = false
  Forces redraw in UI.

Private Methods

- void `addMenu` ()
  Setup the menu bars.

- void `addToolBar` ()
  Setup the toolbars.

- JMenuItem `addItem` (JMenu menu, Action action)
  Adds one menu item.

- JButton `addToolBarButton` (JToolBar toolbar, Action action)
  Adds one toolbar button.

- JButton `addToolBarButton` (JToolBar toolbar, Action action, String icon)
  Adds one toolbar button.
Private Attributes

- JMenuBar menubar = new JMenuBar()
  
  *Bar for command menus.*

- JToolBar toolbar = new JToolBar()

  *Bar for command buttons.*

- StatusBar statusbar = new StatusBar()

  *Displays cur. slice.*

- JMenu filemenu = new JMenu("File")

  *File manipulation menu.*

- JMenu editmenu = new JMenu("Edit")

  *Options editing menu.*

- JMenu viewmenu = new JMenu("View")

  *View change menu.*

- JMenu helpmenu = new JMenu("Help")

  *Reference & help menu.*

- int dCount = 0

  *Determines refreshes for rendered views.*

- int dCountMarkers = 0

  *Redraw only the 2D windows when the selection markers might change.*

- Vector objListReference = new Vector()

  *This vector acts as a reference to a PickablePolygon array representing the polygons currently visible in the ControlWindow (p. 18).*

- MemiHandler newmenu
- MemiHandler openmenu
- MemiHandler closemenu
- MemiHandler savemenu
- MemiHandler saveasmenu
- MemiHandler printmenu
- MemiHandler exitmenu
- MemiHandler nextmenu
- MemiHandler prevmenu
7.1 win.ControlWindow Class Reference

- MenuHandler new2dview
- MenuHandler new3dview
- MenuHandler colormenu
- MenuHandler manual
- MenuHandler license
- MenuHandler aboutmenu
- MenuHandler newsphere
- MenuHandler newcone
- MenuHandler newcylinder
- MenuHandler newcube
- MenuHandler setcolor
- MenuHandler translate
- MenuHandler rotateX
- MenuHandler rotateY
- MenuHandler rotateZ
- MenuHandler scale
- MenuHandler importmenu
- MenuHandler optionsmenu
- MenuHandler delete

7.1.1 Detailed Description

Main view a user will interact with.
Definition at line 51 of file ControlWindow.java.

7.1.2 Member Function Documentation

7.1.2.1 void win.ControlWindow.addMenu () [inline, private]

Setup the menu bars.
Definition at line 271 of file ControlWindow.java.


Referenced by win.ControlWindow.showWindow().

7.1.2.2 JMenuItem win.ControlWindow.addMenuItem (JMenu menu, Action action) [inline, private]

Adds one menu item.
Parameters:
  *menu* menu to add to
  *action* which action to call

Returns:
  Menu item added to menu.

Definition at line 439 of file ControlWindow.java.
Referenced by win.ControlWindow.addMenu().

### 7.1.2.3 void win.ControlWindow.addToolBar () [inline, private]

Setup the toolbars.
Definition at line 392 of file ControlWindow.java.
Referenced by win.ControlWindow.addMenu().

### 7.1.2.4 JButton win.ControlWindow.addToolBarButton (JToolBar toolbar, Action action, String icon) [inline, private]

Adds one toolbar button.

Parameters:
  *toolbar* destination
  *action* action to use
  *icon* picture to use

Returns:
  button created

Definition at line 467 of file ControlWindow.java.
References win.ControlWindow.toolbar.

### 7.1.2.5 JButton win.ControlWindow.addToolBarButton (JToolBar toolbar, Action action) [inline, private]

Adds one toolbar button.

Parameters:
  *toolbar* destination
7.1 win.ControlWindow Class Reference

**action** action to use

**Returns:**

button created

Definition at line 455 of file ControlWindow.java.
References win.ControlWindow.toolbar.
Referenced by win.ControlWindow.addToolBar().

7.1.2.6 void win.ControlWindow.drawFrame (Graphics g, Rectangle r, Render rv, Vector objRef) [inline, static]

Render a frame in a window.

**Parameters:**

- **g** device to draw on
- **r** target view
- **rv** render options to use

Definition at line 262 of file ControlWindow.java.
Referenced by win.ControlWindow.print().

7.1.2.7 int win.ControlWindow.getNumberPages () [inline]

Returns the number of pages to print.
Implemented for interface Pageable.

**Returns:**

Number of pages, or Unknown if not available.

Definition at line 225 of file ControlWindow.java.

7.1.2.8 PageFormat win.ControlWindow.getPageFormat (int page-Index) [inline]

Gets the format for a page.
Implemented for interface Pageable.

**Parameters:**

- **pageIndex** current page number

**Returns:**

Page orientation landscape.

Definition at line 238 of file ControlWindow.java.
7.1.2.9 Printable win.ControlWindow.printable (int pageIndex) [inline]

Gets the object responsible for printing a page.
Implemented for interface Pageable.

Parameters:
pageIndex current page number

Returns:
Current object.

Definition at line 251 of file ControlWindow.java.

7.1.2.10 void win.ControlWindow.hide () [inline]

Hide window when needed.
Definition at line 131 of file ControlWindow.java.

7.1.2.11 int win.ControlWindow.print (Graphics g, PageFormat pageFormat, int pageIndex) [inline]

Print on demand.
Used to implement the Interface Printable.

Parameters:
g device to accept graphics
pageFormat format of the current page
pageIndex current page number

Returns:
PAGE_EXISTS or NO_SUCH_PAGE

Definition at line 206 of file ControlWindow.java.
References win.ControlWindow.drawLine(), win.ControlWindow.objList-
Reference, and win.ControlWindow.view.

7.1.2.12 void win.ControlWindow.showWindow () [inline]

Setup the window, then display it.
Definition at line 90 of file ControlWindow.java.

7.1.2.13 `void win.ControlWindow.updateWindow()` [inline]
Redraw window when needed.
Definition at line 140 of file ControlWindow.java.

7.1.3 Member Data Documentation

7.1.3.1 `MenuHandler win.ControlWindow.aboutmenu` [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.2 `MenuHandler win.ControlWindow.closemenu` [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.3 `MenuHandler win.ControlWindow.colormenu` [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.4 `int win.ControlWindow.dCount = 0` [private]
Determines refreshes for rendered views.
Definition at line 72 of file ControlWindow.java.
Referenced by `win.ControlWindow.updateWindow()`.

7.1.3.5 `int win.ControlWindow.dCountMarkers = 0` [private]
Redraw only the 2D windows when the selection markers might change.
Definition at line 73 of file ControlWindow.java.
Referenced by `win.ControlWindow.updateWindow()`.
7.1.3.6 MenuHandler win.ControlWindow.delete [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.7 boolean win.ControlWindow.dirty = false
Forces redraw in UI.
Definition at line 71 of file ControlWindow.java.
Referenced by win.MenuHandler.actionPerformed(), and win.ControlWindow.updateWindow().

7.1.3.8 JMenuItem win.ControlWindow.editmenu = new JMenuItem("Edit") [private]
Options editing menu.
Definition at line 58 of file ControlWindow.java.
Referenced by win.ControlWindow.addMenu().

7.1.3.9 MenuHandler win.ControlWindow.exitmenu [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.10 JMenuItem win.ControlWindow.filemenu = new JMenuItem("File") [private]
File manipulation menu.
Definition at line 57 of file ControlWindow.java.
Referenced by win.ControlWindow.addMenu().

7.1.3.11 JMenuItem win.ControlWindow.helpmenu = new JMenuItem("Help") [private]
Reference & help menu.
Definition at line 60 of file ControlWindow.java.
Referenced by win.ControlWindow.addMenu().

7.1.3.12 MenuHandler win.ControlWindow.importmenu [private]
Definition at line 63 of file ControlWindow.java.
7.1.3.13 MenuHandler win.ControlWindow.license [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.14 MenuHandler win.ControlWindow.manual [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.15 JMenuBar win.ControlWindow.menuBar = new JMenuBar() [private]
Bar for command menus.
Definition at line 54 of file ControlWindow.java.
Referenced by win.ControlWindow.addMenu().

7.1.3.16 MenuHandler win.ControlWindow.new2dview [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.17 MenuHandler win.ControlWindow.new3dview [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.18 MenuHandler win.ControlWindow.newcone [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.19 MenuHandler win.ControlWindow.newcube [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.20 MenuHandler win.ControlWindow.newcylinder [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.21 MenuHandler win.ControlWindow.newmenu [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.22 MenuHandler win.ControlWindow.newsphere [private]
Definition at line 63 of file ControlWindow.java.

---

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.1.3.23 MenuHandler win.ControlWindow.nextmenu [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.24 Vector win.ControlWindow.objListReference = new Vector() [private]
This vector acts as a reference to a PickablePolygon array representing the polygons currently visible in the ControlWindow (p.18).
Because this array must be set from a static context in the drawFrame() (p. 24) method, we need a reference to the data. Thus, objListReference will always contain exactly one element, and the element will be a PickablePolygon array.
Definition at line 78 of file ControlWindow.java.
Referenced by win.ControlWindow.print(), and win.ControlWindow.showWindow().

7.1.3.25 MenuHandler win.ControlWindow.openmenu [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.26 MenuHandler win.ControlWindow.optionsmenu [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.27 MenuHandler win.ControlWindow.prevmenu [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.28 MenuHandler win.ControlWindow.printmenu [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.29 MenuHandler win.ControlWindow.rotateX [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.30 MenuHandler win.ControlWindow.rotateY [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.31 MenuHandler win.ControlWindow.rotateZ [private]
Definition at line 63 of file ControlWindow.java.
7.1.3.32 MenuHandler win.ControlWindow.saveasmenu [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.33 MenuHandler win.ControlWindow.savemenu [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.34 MenuHandler win.ControlWindow.scale [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.35 MenuHandler win.ControlWindow.setcolor [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.36 StatusBar win.ControlWindow.statusbar = new StatusBar() [private]
Displays cur. slice.
Definition at line 56 of file ControlWindow.java.

7.1.3.37 JToolBar win.ControlWindow.toolbar = new JToolBar() [private]
Bar for command buttons.
Definition at line 55 of file ControlWindow.java.
Referenced by win.ControlWindow.addToolBar(), and win.ControlWindow.addToolBarButton().

7.1.3.38 MenuHandler win.ControlWindow.translate [private]
Definition at line 63 of file ControlWindow.java.

7.1.3.39 Render win.ControlWindow,view
Section of ControlWindow (p.18) where slices are drawn.
Definition at line 70 of file ControlWindow.java.
7.2 win.Display Class Reference

7.1.3.40 JMenu win.ControlWindow.viewMenu = new JMenu("View") [private]

View change menu.
Definition at line 59 of file ControlWindow.java.
Referenced by win.ControlWindow.addMenu().
The documentation for this class was generated from the following file:

- ControlWindow.java

7.2 win.Display Class Reference

Manages 2D and 3D windows in UI.

Public Methods

- Display ()
  Open new display with one 2D and one 3D window.

Static Public Methods

- void newSlice ()
  Open another 2D window.

- void new3D ()
  Open another 3D window.

- void remove2D (ControlWindow win)
  Remove a single 2D window.

- void remove3D (J3DWindow win)
  Remove a single 3D window.

- void updateWindow ()
  Update all 2D and 3D windows.

Static Public Attributes

- LinkedList windowList = new LinkedList()
List of open 2D windows.

- LinkedList **java3DList** = new LinkedList()
  
  List of open 3D windows.

### 7.2.1 Detailed Description

Manages 2D and 3D windows in UI. Definition at line 50 of file Display.java.

### 7.2.2 Constructor & Destructor Documentation

#### 7.2.2.1 `win.Display.Display()` [inline]

Open new display with one 2D and one 3D window. Definition at line 58 of file Display.java. References `win.Display.new3D()`, and `win.Display.newSlice()`.

### 7.2.3 Member Function Documentation

#### 7.2.3.1 `void win.Display.new3D()` [inline, static]


#### 7.2.3.2 `void win.Display.newSlice()` [inline, static]


#### 7.2.3.3 `void win.Display.remove2D (ControlWindow win)` [inline, static]

Remove a single 2D window.
Parameters:

*win* Window to remove

Definition at line 97 of file Display.java.
References win.Display.newSlice(), and win.Display.windowList.

### 7.2.3.4 void win,Display.remove3D (J3DWindow win) [inline, static]
Remove a single 3D window.

Parameters:

*win* Window to remove

Definition at line 120 of file Display.java.
References win.Display.java3DList.

### 7.2.3.5 void win,Display.updateWindow () [inline, static]
Update all 2D and 3D windows.

Definition at line 133 of file Display.java.
References win.Display.java3DList, win.J3DWindow.updateWindow(),

### 7.2.4 Member Data Documentation

#### 7.2.4.1 LinkedList win,Display.java3DList = new LinkedList() [static]
List of open 3D windows.

Definition at line 53 of file Display.java.
Referenced by win.Display.new3D(), win.Display.remove3D(), and
win.Display.updateWindow().

#### 7.2.4.2 LinkedList win,Display.windowList = new LinkedList() [static]
List of open 2D windows.

Definition at line 52 of file Display.java.
Referenced by win.Display.newSlice(), win.Display.remove2D(), and
win.Display.updateWindow().

The documentation for this class was generated from the following file:

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.3 win.ExtensionFilter Class Reference

Handles extension options in a save/open dialog box. Inheritance diagram for win.ExtensionFilter:

![Inheritance Diagram]

Collaboration diagram for win.ExtensionFilter:

![Collaboration Diagram]

Public Methods

- **ExtensionFilter** (String ext, String descr)
  
  Default constructor.

- boolean **accept** (File file)
  
  Decide if a file matches the filter.

- String **getDescription** ()
  
  Get the description of the filter.

Private Attributes

- String **description**
  
  Description of the extension.

- String **extension**
  
  Textual composition of extension.
7.3.1 Detailed Description

Handles extension options in a save/open dialog box.
Definition at line 41 of file ExtensionFilter.java.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 win.ExtensionFilter.ExtensionFilter (String ext, String descr) [inline]
Default constructor.

Parameters:
  ext Actual extension
  descr Description of extension

Definition at line 48 of file ExtensionFilter.java.

7.3.3 Member Function Documentation

7.3.3.1 boolean win.ExtensionFilter.accept (File file) [inline]
Decide if a file matches the filter.

Parameters:
  file filename to check

Returns:
  is it allowed

Definition at line 59 of file ExtensionFilter.java.
References win.ExtensionFilter.extension.

7.3.3.2 String win.ExtensionFilter.getDescription () [inline]
Get the description of the filter.

Returns:
  the description

Definition at line 69 of file ExtensionFilter.java.
References win.ExtensionFilter.description.
7.3.4 Member Data Documentation

7.3.4.1 String win.ExtensionFilter.description [private]
Description of the extension.
Definition at line 73 of file ExtensionFilter.java.
Referenced by win.ExtensionFilter.ExtensionFilter(), and win.ExtensionFilter.getDescription().

7.3.4.2 String win.ExtensionFilter.extension [private]
Textual composition of extension.
Definition at line 74 of file ExtensionFilter.java.
Referenced by win.ExtensionFilter.accept(), and win.ExtensionFilter.ExtensionFilter().
The documentation for this class was generated from the following file:

- ExtensionFilter.java

7.4 disp.Geometry Class Reference

Static class to handle conversion from inches to pixels.

Static Public Methods

- double inch2pixX (double inchesX)
  Convert from a Cartesian x position in inches to pixels.

- double inch2pixY (double inchesY)
  Convert from a Cartesian y position in inches to pixels.

- double inch2distX (double inchesX)
  Get the distance in pixels.

- double inch2distY (double inchesY)
  Get the distance in pixels.

- void setTranslation (Graphics2D g, Rectangle r)
  Setup for the translations.

- double layer2pixZ (int layer, int numLayers)
Find the z height for a specific layer.

- double `inch2meter` (double inch)
  
  Convert inches to meters.

- double `getxMin` ()
- double `getxMax` ()
- double `getyMin` ()
- double `getyMax` ()
- double `getzMin` ()
- double `getzMax` ()
- void `setzMin` (double z)
- void `setzMax` (double z)

**Static Public Attributes**

- boolean `Printing`
  
  Set where we need conversion for printing.

**Static Private Attributes**

- Rectangle `target`
  
  Destination to map onto.

- double `xMin` = -4.5
  
  Default box parameter.

- double `xMax` = 4.5
  
  Default box parameter.

- double `yMin` = -3.25
  
  Default box parameter.

- double `yMax` = 3.25
  
  Default box parameter.

- double `zMin` = -3.75
  
  Default box parameter.

- double `zMax` = 3.75
  
  Default box parameter.
7.4.1 Detailed Description

Static class to handle conversion from inches to pixels.
Definition at line 38 of file Geometry.java.

7.4.2 Member Function Documentation

7.4.2.1 double disp.Geometry.getxMax () [inline, static]
Definition at line 137 of file Geometry.java.
References disp.Geometry.xMax.

7.4.2.2 double disp.Geometry.getxMin () [inline, static]
Definition at line 134 of file Geometry.java.
References disp.Geometry.xMin.

7.4.2.3 double disp.Geometry.getyMax () [inline, static]
Definition at line 143 of file Geometry.java.
References disp.Geometry.yMax.

7.4.2.4 double disp.Geometry.getyMin () [inline, static]
Definition at line 140 of file Geometry.java.
References disp.Geometry.yMin.

7.4.2.5 double disp.Geometry.getzMax () [inline, static]
Definition at line 149 of file Geometry.java.
References disp.Geometry.zMax.

7.4.2.6 double disp.Geometry.getzMin () [inline, static]
Definition at line 146 of file Geometry.java.
References disp.Geometry.zMin.

7.4.2.7 double disp.Geometry.inch2distX (double inchesX) [inline, static]
Get the distance in pixels.

---

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
Parameters:
   \textit{inchesX} length in inches

Returns :
   length in pixels

Definition at line 73 of file Geometry.java.

7.4.2.8 double disp.Geometry.inch2distY (double \textit{inchesY})
   [inline, static]
Get the distance in pixels.

Parameters:
   \textit{inchesY} length in inches

Returns :
   length in pixels

Definition at line 83 of file Geometry.java.

7.4.2.9 double disp.Geometry.inch2meter (double \textit{inch})
   [inline, static]
Convert inches to meters.
This is useful for Java3D, which uses meters for all units.

Parameters:
   \textit{inch} Distance in inches

Returns :
   Distance in meters

Definition at line 120 of file Geometry.java.

7.4.2.10 double disp.Geometry.inch2pixX (double \textit{inchesX})
   [inline, static]
Convert from a Cartesian x position in inches to pixels.
Suitable for drawing on a graphics object conforming to the geometry specifications defined in this object.
Parameters:
  \textit{inchesX} \ x \ offset

Returns:
  offset in pixels

Definition at line 47 of file Geometry.java.
References \texttt{disp.Geometry.target}, \texttt{disp.Geometry.xMax}, and \texttt{disp.Geometry.xMin}.

7.4.2.11 \texttt{double disp.Geometry.inch2pixY (double inchesY)}
\texttt{[inline, static]}
Convert from a Cartesian \(y\) position in inches to pixels.
Suitable for drawing on a graphics object conforming to the geometry specifications defined in this object.

Parameters:
  \textit{inchesY} \ y \ offset \ in \ inches

Returns:
  offset in pixels

Definition at line 59 of file Geometry.java.

7.4.2.12 \texttt{double disp.Geometry.layer2pixZ (int layer, int numLayers)}
\texttt{[inline, static]}
Find the \(z\) height for a specific layer.

Parameters:
  \textit{layer} \ current \ layer
  \textit{numLayers} \ max \ layer

Returns:
  \(z\) \ height

Definition at line 109 of file Geometry.java.
References \texttt{disp.Geometry.zMax}, and \texttt{disp.Geometry.zMin}.
7.4.2.13 void disp.Geometry.setTranslation (Graphics2D g, Rectangle r) [inline, static]
Setup for the translations.
This must be called before inch2*.

Parameters:
  g currently unused
  r used to store translations

Definition at line 93 of file Geometry.java.
References disp.Geometry.target.

7.4.2.14 void disp.Geometry.setzMax (double z) [inline, static]

Definition at line 155 of file Geometry.java.
References disp.Geometry.zMax.

7.4.2.15 void disp.Geometry.setzMin (double z) [inline, static]

Definition at line 152 of file Geometry.java.
References disp.Geometry.zMin.

7.4.3 Member Data Documentation

7.4.3.1 boolean disp.Geometry.Printing [static]
Set where we need conversion for printing.
Definition at line 162 of file Geometry.java.

7.4.3.2 Rectangle disp.Geometry.target [static, private]
Destination to map onto.
Definition at line 125 of file Geometry.java.

7.4.3.3 double disp.Geometry.xMax = 4.5 [static, private]
Default box parameter.
7.4 disp.Geometry Class Reference

Definition at line 128 of file Geometry.java.

7.4.3.4 double disp.Geometry.xMin = -4.5 [static, private]
Default box parameter.
Definition at line 127 of file Geometry.java.

7.4.3.5 double disp.Geometry.yMax = 3.25 [static, private]
Default box parameter.
Definition at line 130 of file Geometry.java.

7.4.3.6 double disp.Geometry.yMin = -3.25 [static, private]
Default box parameter.
Definition at line 129 of file Geometry.java.

7.4.3.7 double disp.Geometry.zMax = 3.75 [static, private]
Default box parameter.
Definition at line 132 of file Geometry.java.
Referenced by disp.Geometry.getZMax(), disp.Geometry.layer2pixZ(), and disp.Geometry.setzMax().

7.4.3.8 double disp.Geometry.zMin = -3.75 [static, private]
Default box parameter.
Definition at line 131 of file Geometry.java.

The documentation for this class was generated from the following file:

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.5 win.J3DWindow Class Reference

3D view of a scene.

Public Methods

- `void showWindow ()`
  
  Setup the window, then display it.

- `void hide ()`
  
  Hides the 3D window.

- `void updateWindow ()`
  
  Used to let the window know some things may have changed.

- `BranchGroup getCurrentScene ()`
  
  Return a BranchGroup describing the current scene suitable for use in a Java3D universe.

Private Attributes

- `Canvas3D canvas3D = null`
  
  Canvas to draw the 3D objects on.

- `int dCount = 0`
  
  Used to keep track of whether scene needs to be updated.

- `TransformGroup sceneRot = null`
  
  Topmost node of the scenegraph.

7.5.1 Detailed Description

3D view of a scene.

Definition at line 47 of file J3DWindow.java.

---

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.5.2 Member Function Documentation

7.5.2.1 BranchGroup win.J3DWindow.getCurrentScene () [inline]
Return a BranchGroup describing the current scene suitable for use in a Java3D universe.

Returns:
Returns the current 3D scene

Definition at line 159 of file J3DWindow.java.
Referenced by win.J3DWindow.updateWindow().

7.5.2.2 void win.J3DWindow.hide () [inline]
Hides the 3D window.
Definition at line 124 of file J3DWindow.java.

7.5.2.3 void win.J3DWindow.showWindow () [inline]
Setup the window, then display it.
Definition at line 57 of file J3DWindow.java.
Referenced by win.Display.new3D().

7.5.2.4 void win.J3DWindow.updateWindow () [inline]
Used to let the window know some things may have changed.
Definition at line 133 of file J3DWindow.java.
Referenced by win.J3DWindow.showWindow(), and win.Display.updateWindow().

7.5.3 Member Data Documentation

7.5.3.1 Canvas3D win.J3DWindow.canvas3D = null [private]
Canvas to draw the 3D objects on.
Definition at line 48 of file J3DWindow.java.
Referenced by win.J3DWindow.showWindow().

7.5.3.2 int win.J3DWindow.dCount = 0 [private]
Used to keep track of whether scene needs to be updated.
Definition at line 49 of file J3DWindow.java.
Referenced by win.J3DWindow.updateWindow().

7.5.3.3 TransformGroup win.J3DWindow.sceneRot = null [private]
Topmost node of the scenegraph.
Definition at line 51 of file J3DWindow.java.
Referenced by win.J3DWindow.showWindow(), and win.J3DWindow.updateWindow().
The documentation for this class was generated from the following file:

- J3DWindow.java

7.6 win.MainWindow Class Reference

This does nothing other than open up the control window.
Inheritance diagram for win.MainWindow:

```
      JApplet
     /   
win.MainWindow
```

Collaboration diagram for win.MainWindow:

```
      JApplet
     /   
win.MainWindow  window
```
Public Methods

- void init ()
  
  This is called from main, or when the applet is loaded.

Static Public Methods

- void main (String[] args)

  The starting place for the program.

Static Public Attributes

- Main Window window

  This is the main window if not run as applet.

- boolean runAsApplet = true

  This tells whether it is run as applet.

- int majorVersion = 1

  Major version number for SPECTRE.

- int minorVersion = 0

  Minor version number for SPECTRE.

7.6.1 Detailed Description

This does nothing other than open up the control window.
Definition at line 40 of file MainWindow.java.

7.6.2 Member Function Documentation

7.6.2.1 void win.MainWindow.init () [inline]

This is called from main, or when the applet is loaded.
Definition at line 64 of file MainWindow.java.
Referenced by win.MainWindow.main().
7.6.2.2 void win.MainWindow.main (String args[]) [inline, static]
The starting place for the program.

Parameters:
   args anything passed on command line (ignored)

Definition at line 55 of file MainWindow.java.
References win.MainWindow.init(), and win.MainWindow.runAsApplet.

7.6.3 Member Data Documentation

7.6.3.1 int win.MainWindow.majorVersion = 1 [static]
Major version number for SPECTRE.
Definition at line 47 of file MainWindow.java.

7.6.3.2 int win.MainWindow.minorVersion = 0 [static]
Minor version number for SPECTRE.
Definition at line 48 of file MainWindow.java.

7.6.3.3 boolean win.MainWindow.runAsApplet = true [static]
This tells whether it is run as applet.
Definition at line 44 of file MainWindow.java.
Referenced by win.MainWindow.main().

7.6.3.4 MainWindow win.MainWindow.window [static]
This is the main window if not run as applet.
Definition at line 41 of file MainWindow.java.
The documentation for this class was generated from the following file:

- MainWindow.java

7.7 win.MenuHandler Class Reference

Handles menu accesses.
Collaboration diagram for win.MenuHandler:
Public Methods

- **MenuHandler** (ControlWindow cb, String name, String tooltip)
  
  *Constructor for MenuHandler* (p. 47).

- **MenuHandler** (ControlWindow cb, String name, KeyStroke ks, String tooltip)
  
  *Constructor for MenuHandler* (p. 47).

- void **actionPerformed** (ActionEvent e)
  
  *Handles menu clicks.*

Static Public Methods

- boolean **saveScenePrompt** ()
  
  *Prompts to decide if saving is necessary.*
Static Public Attributes

- Parser parser

Private Methods

- void loadScene ()
  
  Load in a scene from the disk.

Static Private Methods

- boolean saveScene (boolean saveAs)
  
  Saves the current scene.

- File showDialog (String dialogTitle, String approveText, String approveTooltip, char approveMnemonic, File file)
  
  Helps with showing the file chooser dialog.

Private Attributes

- ControlWindow callback
  
  Handles menu and toolbar callbacks.

Static Private Attributes

- JFileChooser files = new JFileChooser("example")
  
  Used for loading files.

7.7.1 Detailed Description

Handles menu accesses.

Definition at line 57 of file MenuHandler.java.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 win.MenuHandler.MenuHandler (ControlWindow cb, String name, String tooltip) [inline]
Constructor for MenuHandler (p. 47).

**Parameters:**
- `cb` Callback for menu item
- `name` text displayed in menu
- `tooltip` text displayed on hover

Definition at line 69 of file MenuHandler.java.

7.7.2.2  win.MenuHandler.MenuHandler (ControlWindow cb, String name, KeyStroke ks, String tooltip)  [inline]
Constructor for MenuHandler (p. 47).

**Parameters:**
- `cb` Callback for menu item
- `name` text displayed in menu
- `ks` shortcut key
- `tooltip` text displayed on hover

Definition at line 84 of file MenuHandler.java.

7.7.3  Member Function Documentation

7.7.3.1  void win.MenuHandler.actionPerformed (ActionEvent e)  [inline]
Handles menu clicks.

**Parameters:**
- `e` action to take

Definition at line 95 of file MenuHandler.java.


7.7.3.2  void win.MenuHandler.loadScene ()  [inline, private]
Load in a scene from the disk.
Definition at line 314 of file MenuHandler.java.

Referenced by win.MenuHandler.actionPerformed().

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.7.3.3 boolean win.MenuHandler.saveScene (boolean saveAs) [inline, static, private]
Saves the current scene.

Parameters:
   saveAs does the user wish a rename

Returns:
   true if saved, false otherwise

Definition at line 262 of file MenuHandler.java.
Referenced by win.MenuHandler.actionPerformed().

7.7.3.4 boolean win.MenuHandler.saveScenePrompt () [inline, static]
Prompts to decide if saving is necessary.

Returns:
   true if can exit, false otherwise

Definition at line 232 of file MenuHandler.java.
Referenced by win.MenuHandler.actionPerformed().

7.7.3.5 File win.MenuHandler.showDialog (String dialogTitle, String approveText, String approveTooltip, char approveMnemonic, File file) [inline, static, private]
Helps with showing the file chooser dialog.

Parameters:
   dialogTitle string for the title bar
   approveText string for the button
   approveTooltip string for the tooltip on the approve button
   approveMnemonic character shortcut
   file default file

Definition at line 393 of file MenuHandler.java.

7.7.4 Member Data Documentation

7.7.4.1 ControlWindow win.MenuHandler.callback [private]
Handles menu and toolbar callbacks.
Definition at line 59 of file MenuHandler.java.
7.7.4.2 `FileChooser` `win.MenuHandler.files = newFileChooser("example")` [static, private]
Used for loading files.
Definition at line 414 of file MenuHandler.java.

7.7.4.3 `Parser` `win.MenuHandler.parser` [static]
Definition at line 60 of file MenuHandler.java.
The documentation for this class was generated from the following file:

- `MenuHandler.java`

7.8 `win.ObjDialog Class Reference`
Creates a multi-use dialog box.

Static Public Methods
- `void resetDefaults ()`
  
  Reset the defaults for the dialog boxes.

- `void showDialog (final String title, ControlWindow callback, final Scene s)`
  
  This function creates and shows a dialog box.

Static Public Attributes
- `JDialog dialog`
  
  Dialog box to manipulate.

- `JPanel Hpanel = new JPanel()`
  
  Height input panel.

- `JPanel Wpanel = new JPanel()`
  
  Width input panel.

- `JPanel Lpanel = new JPanel()`
  
  Length input panel.

- `JPanel Rpanel = new JPanel()`
Radius input panel

- JPanel ObjXpanel = new JPanel()
  X-coord input panel.
- JPanel ObjYpanel = new JPanel()
  Y-coord input panel.
- JPanel ObjZpanel = new JPanel()
  Z-coord input panel.
- JPanel SclXpanel = new JPanel()
  X-coord input panel.
- JPanel SclYpanel = new JPanel()
  Y-coord input panel.
- JPanel SclZpanel = new JPanel()
  Z-coord input panel.
- JPanel TranXpanel = new JPanel()
  X-coord input panel.
- JPanel TranYpanel = new JPanel()
  Y-coord input panel.
- JPanel TranZpanel = new JPanel()
  Z-coord input panel.
- JPanel Spanel = new JPanel()
  Scale factor input panel.
- JTextField Htext = new JTextField("1", 5)
  Height input field.
- JTextField Wtext = new JTextField("1", 5)
  Width input field.
- JTextField Ltext = new JTextField("1", 5)
  Length input field.
- JTextField Rtext = new JTextField("1", 5)
Radius input field.

* JTextField ObjXtext = new JTextField("0", 5)
  X-coord input field.

* JTextField ObjYtext = new JTextField("0", 5)
  Y-coord input field.

* JTextField ObjZtext = new JTextField("0", 5)
  Z-coord input field.

* JTextField Sclxtext = new JTextField("0", 5)
  X-coord input field.

* JTextField SclYtext = new JTextField("0", 5)
  Y-coord input field.

* JTextField SclZtext = new JTextField("0", 5)
  Z-coord input field.

* JTextField TranXtext = new JTextField("0", 5)
  X-coord input field.

* JTextField TranYtext = new JTextField("0", 5)
  Y-coord input field.

* JTextField TranZtext = new JTextField("0", 5)
  Z-coord input field.

* JTextField Stext = new JTextField("0", 5)
  Scale input field.

* JLabel Hlabel = new JLabel("Height")
  Height field label.

* JLabel Wlabel = new JLabel("Width")
  Width field label.

* JLabel Llabel = new JLabel("Length")
  Length field label.

* JLabel Rlabel = new JLabel("Radius")
Radius field label

- JLabel ObjXlabel = new JLabel("X")
  X-coord field label
- JLabel ObjYlabel = new JLabel("Y")
  Y-coord field label
- JLabel ObjZlabel = new JLabel("Z")
  Z-coord field label
- JLabel SclXlabel = new JLabel("X")
  X-coord field label
- JLabel SclYlabel = new JLabel("Y")
  Y-coord field label
- JLabel SclZlabel = new JLabel("Z")
  Z-coord field label
- JLabel TranXlabel = new JLabel("X")
  X-coord field label
- JLabel TranYlabel = new JLabel("Y")
  Y-coord field label
- JLabel TranZlabel = new JLabel("Z")
  Z-coord field label
- JLabel Slabel = new JLabel("Degrees")
  Scale factor field label
- JLabel Olabel = new JLabel("Rotate about: ")
- JRadioButton OriginButton = new JRadioButton("Origin")
- JRadioButton CenterButton = new JRadioButton("Center", true)
- ButtonGroup OriginCenter = new ButtonGroup()
- FocusListener autoHighlight

- double height
- double width
- double length
- double radius
- double degrees
• double Objx
• double Objy
• double Objz
• double Sclx
• double Scly
• double Sclz
• double Trans
• double Trany
• double Trans

Static Private Methods

• void setPanels ()
  
  Sets up the panels for the dialog box.

• boolean readInput (String title)
  
  Converts the values in the text boxes to doubles.

7.8.1 Detailed Description

Creates a multi-use dialog box.
Definition at line 65 of file ObjDialog.java.

7.8.2 Member Function Documentation

7.8.2.1 boolean win.ObjDialog.readInput (String title) [inline, static, private]

Converts the values in the text boxes to doubles.
Definition at line 527 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.2.2 void win ObjDialog.resetDefaults () [inline, static]

Reset the defaults for the dialog boxes.
Definition at line 135 of file ObjDialog.java.
7.8.2.3 void win._ObjDialog.setPanels () [inline, static, private]

Sets up the panels for the dialog box.
Definition at line 480 of file ObjDialog.java.
Referenced by win._ObjDialog.showDialog().

7.8.2.4 void win._ObjDialog.showDialog (final String title, Control-Window callback, final Scene s) [inline, static]

This function creates and shows a dialog box.

Parameters:
    title name & type of dialog box
    callback parent window of this dialog
    s scene in which to perform action

Definition at line 152 of file ObjDialog.java.

7.8.3 Member Data Documentation

7.8.3.1 FocusListener win._ObjDialog.autoHighlight [static]

Initial value:

```java
new FocusAdapter()
{
    public void focusGained(FocusEvent e) {
```
((JTextComponent) e.getComponent()).selectAll();
}
}

Definition at line 126 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.2 JRadioButton win.ObjDialog.CenterButton = new JRadioButton("Center", true) [static]
Definition at line 122 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.3 double win.ObjDialog.degrees [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.4 JDialog win.ObjDialog.dialog [static]
Dialog box to manipulate.
Definition at line 66 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.5 double win.ObjDialog.height [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.6 JLabel win.ObjDialog.Hlabel = new JLabel("Height") [static]
Height field label.
Definition at line 105 of file ObjDialog.java.

7.8.3.7 JPanel win.ObjDialog.Hpanel = new JPanel() [static]
Height input panel.
Definition at line 74 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.8.3.8 JTextField win.ObjDialog.Htext = new JTextField(”1”, 5) [static]
Height input field.
Definition at line 90 of file ObjDialog.java.
Referenced by win.ObjDialog.resetDefaults(), and win.ObjDialog.showDialog().

7.8.3.9 double win.ObjDialog.length [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.10 JLabel win.ObjDialog.Llabel = new JLabel(”Length”) [static]
Length field label.
Definition at line 107 of file ObjDialog.java.

7.8.3.11 JPanel win.ObjDialog.Lpanel = new JPanel() [static]
Length input panel.
Definition at line 76 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.12 JTextField win.ObjDialog.Ltext = new JTextField(”1”, 5) [static]
Length input field.
Definition at line 92 of file ObjDialog.java.
Referenced by win.ObjDialog.resetDefaults(), and win.ObjDialog.showDialog().

7.8.3.13 double win.ObjDialog.Objx [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.14 JLabel win.ObjDialog.ObjXlabel = new JLabel(”X”) [static]
X-coord field label.
Definition at line 109 of file ObjDialog.java.

7.8.3.15  
```
JPanel win,ObjDialog,ObjXpanel = new JPanel()
```
[static]
X-coord input panel.
Definition at line 78 of file ObjDialog.java.
Referenced by win,ObjDialog.showDialog().

7.8.3.16  
```
JTextField win,ObjDialog,ObjXtext = new JTextField("0", 5) [static]
```
X-coord input field.
Definition at line 94 of file ObjDialog.java.
Referenced by win,ObjDialog.resetDefaults(), and win,ObjDialog.showDialog().

7.8.3.17  
```
double win,ObjDialog,Objy [static]
```
Definition at line 70 of file ObjDialog.java.
Referenced by win,ObjDialog.showDialog().

7.8.3.18  
```
JLabel win,ObjDialog,ObjYlabel = new JLabel("Y") [static]
```
Y-coord field label.
Definition at line 110 of file ObjDialog.java.

7.8.3.19  
```
JPanel win,ObjDialog,ObjYpanel = new JPanel() [static]
```
Y-coord input panel.
Definition at line 79 of file ObjDialog.java.
Referenced by win,ObjDialog.showDialog().

7.8.3.20  
```
JTextField win,ObjDialog,ObjYtext = new JTextField("0", 5) [static]
```
Y-coord input field.
Definition at line 95 of file ObjDialog.java.
Referenced by win.ObjDialog.resetDefaults(), and win.ObjDialog.showDialog().

7.8.3.21  double win.ObjDialog.ObjZ  [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.22  JLabel win.ObjDialog.ObjZlabel = new JLabel("Z")  [static]
Z-coord field label.
Definition at line 111 of file ObjDialog.java.

Z-coord input panel.
Definition at line 80 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.24  JTextField win.ObjDialog.ObjZtext = new JTextField("0", 5)  [static]
Z-coord input field.
Definition at line 96 of file ObjDialog.java.
Referenced by win.ObjDialog.resetDefaults(), and win.ObjDialog.showDialog().

7.8.3.25  JLabel win.ObjDialog.Oblabel = new JLabel("Rotate about:")  [static]
Definition at line 119 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

Definition at line 121 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().
7.8.3.27 ButtonGroup win. ObjDialog. OriginCenter = new ButtonGroup() [static]
Definition at line 123 of file ObjDialog.java.
Referenced by win. ObjDialog. showDialog() .

7.8.3.28 double win. ObjDialog. radius [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win. ObjDialog. showDialog() .

7.8.3.29 JLabel win. ObjDialog. Rlabel = new JLabel("Radius") [static]
Radius field label.
Definition at line 108 of file ObjDialog.java.

Radius input panel.
Definition at line 77 of file ObjDialog.java.
Referenced by win. ObjDialog. showDialog() .

7.8.3.31 JTextField win. ObjDialog. Rtext = new JTextField("1", 5) [static]
Radius input field.
Definition at line 93 of file ObjDialog.java.

7.8.3.32 double win. ObjDialog. Sclx [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win. ObjDialog. showDialog() .

7.8.3.33 JLabel win. ObjDialog. SclXlabel = new JLabel("X") [static]
X-coord field label.
Definition at line 112 of file ObjDialog.java.
7.8.3.34 JPanel win.ObjDialog.SclXpanel = new JPanel() [static]

X-coord input panel.
Definition at line 81 of file ObjDialog.java.
Referred by win.ObjDialog.showDialog().

7.8.3.35 JTextField win.ObjDialog.SclXtext = new JTextField(“0”,
5) [static]
X-coord input field.
Definition at line 97 of file ObjDialog.java.
Referred by win.ObjDialog.showDialog().

7.8.3.36 double win.ObjDialog.Scly [static]
Definition at line 70 of file ObjDialog.java.
Referred by win.ObjDialog.showDialog().

7.8.3.37 JLabel win.ObjDialog.SclYlabel = new JLabel(“Y”) [static]
Y-coord field label.
Definition at line 113 of file ObjDialog.java.


Y-coord input panel.
Definition at line 82 of file ObjDialog.java.
Referred by win.ObjDialog.showDialog().

7.8.3.39 JTextField win.ObjDialog.SclYtext = new JTextField(“0”,
5) [static]
Y-coord input field.
Definition at line 98 of file ObjDialog.java.
Referred by win.ObjDialog.showDialog().

7.8.3.40 double win.ObjDialog.Selz [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.41 JLabel win,ObjDialog.SclZlabel = new JLabel("Z")
[static]
Z-coord field label.
Definition at line 114 of file ObjDialog.java.

7.8.3.42 JPanel win,ObjDialog.SclZpanel = new JPanel() [static]
Z-coord input panel.
Definition at line 83 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.43 JTextField win,ObjDialog.SclZtext = new JTextField("0", 5) [static]
Z-coord input field.
Definition at line 99 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.44 JLabel win,ObjDialog.Slabel = new JLabel("Degrees")
[static]
Scale factor field label.
Definition at line 118 of file ObjDialog.java.

7.8.3.45 JPanel win,ObjDialog.Spanel = new JPanel() [static]
Scale factor input panel.
Definition at line 87 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.46 JTextField win,ObjDialog.Stext = new JTextField("0", 5)
[static]
Scale input field.
Definition at line 103 of file ObjDialog.java.

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
Referenced by win.ObjDialog.showDialog().

### 7.8.3.47 double win.ObjDialog.TranX [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

### 7.8.3.48 JLabel win.ObjDialog.TranXlabel = new JLabel("X") [static]
X-coord field label.
Definition at line 115 of file ObjDialog.java.

### 7.8.3.49 JPanel win.ObjDialog.TranXpanel = new JPanel() [static]
X-coord input panel.
Definition at line 84 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

### 7.8.3.50 JTextField win.ObjDialog.TranXtext = new JTextField("0", 5) [static]
X-coord input field.
Definition at line 100 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

### 7.8.3.51 double win.ObjDialog.TranY [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

### 7.8.3.52 JLabel win.ObjDialog.TranYlabel = new JLabel("Y") [static]
Y-coord field label.
Definition at line 116 of file ObjDialog.java.

### 7.8.3.53 JPanel win.ObjDialog.TranYpanel = new JPanel() [static]
7.8 win.ObjDialog Class Reference

Y-coord input panel.
Definition at line 85 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.54 JTextField win.ObjDialog.TranYtext = new JTextField("0", 5) [static]
Y-coord input field.
Definition at line 101 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.55 double win.ObjDialog.Tranz [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.56 JLabel win.ObjDialog.TranZlabel = new JLabel("Z")
Z-coord field label.
Definition at line 117 of file ObjDialog.java.

7.8.3.57 JPanel win.ObjDialog.TranZpanel = new JPanel()
Z-coord input panel.
Definition at line 86 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.58 JTextField win.ObjDialog.TranZtext = new JTextField("0", 5) [static]
Z-coord input field.
Definition at line 102 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.59 double win.ObjDialog.width [static]
Definition at line 70 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().
7.8.3.60 JLabel win.ObjDialog.Wlabel = new JLabel("Width")
[static]
Width field label.
Definition at line 106 of file ObjDialog.java.

7.8.3.61 JPanel win.ObjDialog.Wpanel = new JPanel() [static]
Width input panel.
Definition at line 75 of file ObjDialog.java.
Referenced by win.ObjDialog.showDialog().

7.8.3.62 JTextField win.ObjDialog.Wtext = new JTextField("1", 5)
[static]
Width input field.
Definition at line 91 of file ObjDialog.java.
Referenced by win.ObjDialog.resetDefaults(), and win.ObjDialog.showDialog().
The documentation for this class was generated from the following file:

- ObjDialog.java

7.9 obj.ObjFactory Class Reference

Creates Sphere, Cube, Cone, and Cylinders from Polymeshes.

Collaboration diagram for obj.ObjFactory:

---

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
Static Public Methods

- **Polymesh newCube ()**
  
  Return a unit-cube.

- **Polymesh newCone ()**
  
  Return a unit-cone.

- **Polymesh newSphere ()**
  
  Return a unit-sphere.

- **Polymesh newCylinder ()**
  
  Return a unit-cylinder.

Static Private Attributes

- **Polymesh cubeTessellation = null**
- **Polymesh sphereTessellation = null**
- **Polymesh cylinderTessellation = null**
- **Polymesh coneTessellation = null**

7.9.1 Detailed Description

Creates Sphere, Cube, Cone, and Cylinders from Polymeshes.
Definition at line 34 of file ObjFactory.java.

7.9.2 Member Function Documentation

7.9.2.1 **Polymesh obj.ObjFactory.newCone () [inline, static]**

Return a unit-cone.
The radius and height are 1 unit, centered at the origin, and its point oriented along the z axis.
Definition at line 81 of file ObjFactory.java.

7.9.2.2 **Polymesh obj.ObjFactory.newCube () [inline, static]**

Return a unit-cube.
Each side is 1 unit long, centered at the origin, with the normal of each face parallel to the x, y, or z axis.
Definition at line 44 of file ObjFactory.java.
References Obj.Polymesh.edgeify().

7.9.2.3 Polymesh obj.ObjFactory.newCylinder() [inline, static]

Return a unit-cylinder.
Radius is 1 unit and height is 1 unit. The cylinder is oriented along the z-axis, with its center at the origin.
Definition at line 253 of file ObjFactory.java.

7.9.2.4 Polymesh obj.ObjFactory.newSphere() [inline, static]

Return a unit-sphere.
The radius is 1 unit long, centered at the origin.
Definition at line 131 of file ObjFactory.java.

7.9.3 Member Data Documentation

7.9.3.1 Polymesh obj.ObjFactory.coneTessellation = null [static, private]
Definition at line 38 of file ObjFactory.java.

7.9.3.2 Polymesh obj.ObjFactory.cubeTessellation = null [static, private]
Definition at line 35 of file ObjFactory.java.

7.9.3.3 Polymesh obj.ObjFactory.cylinderTessellation = null [static, private]
Definition at line 37 of file ObjFactory.java.

7.9.3.4 Polymesh obj.ObjFactory.sphereTessellation = null [static, private]
Definition at line 36 of file ObjFactory.java.
The documentation for this class was generated from the following file:

- ObjFactory.java
7.10 win.OptionsDialog Class Reference

Class to display the options dialog.

Static Public Methods

- void `showDialog` (ControlWindow callback)
  
  Show the dialog box.

Static Public Attributes

- JDialog `dialog`
- JSpinner `numSlices` = new JSpinner()
- JSpinner `devHeight` = new JSpinner()
- JCheckBox `drawGrid`
- JCheckBox `printCalib`
- boolean `grid`
- boolean `calib`

7.10.1 Detailed Description

Class to display the options dialog.

Definition at line 46 of file OptionsDialog.java.

7.10.2 Member Function Documentation

7.10.2.1 void `win.OptionsDialog.showDialog` (ControlWindow callback) [inline, static]

Show the dialog box.

Parameters:

  `callback` Calling window. Used to get additional information.

Definition at line 59 of file OptionsDialog.java.


Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.10.3 Member Data Documentation

7.10.3.1 boolean win.OptionsDialog.calib [static]
Definition at line 52 of file OptionsDialog.java.
Referenced by win.OptionsDialog.showDialog().

7.10.3.2 JSpinner win.OptionsDialog.devHeight = new JSpinner() [static]
Definition at line 49 of file OptionsDialog.java.
Referenced by win.OptionsDialog.showDialog().

7.10.3.3 JDialog win.OptionsDialog.dialog [static]
Definition at line 47 of file OptionsDialog.java.
Referenced by win.OptionsDialog.showDialog().

7.10.3.4 JCheckBox win.OptionsDialog.drawGrid [static]
Definition at line 50 of file OptionsDialog.java.
Referenced by win.OptionsDialog.showDialog().

7.10.3.5 boolean win.OptionsDialog.grid [static]
Definition at line 52 of file OptionsDialog.java.
Referenced by win.OptionsDialog.showDialog().

7.10.3.6 JSpinner win.OptionsDialog.numSlices = new JSpinner() [static]
Definition at line 48 of file OptionsDialog.java.
Referenced by win.OptionsDialog.showDialog().

7.10.3.7 JCheckBox win.OptionsDialog.printCalib [static]
Definition at line 51 of file OptionsDialog.java.
Referenced by win.OptionsDialog.showDialog().
The documentation for this class was generated from the following file:

- OptionsDialog.java
7.11 io.ParseException Class Reference

This exception is thrown when parse errors are encountered.

Collaboration diagram for io.ParseException:

```
io.Token  next
         specialToken
         currentToken
         io.ParseException
```

Public Methods

- **ParseException** (Token currentTokenVal, int[][] expectedTokenSequencesVal, String[] tokenImageVal)
  
  *This constructor is used by the method "generateParseException" in the generated parser.*

- **ParseException ()**
  
  *The following constructors are for use by you for whatever purpose you can think of.*

- **ParseException (String message)**
- **String getMessage ()**
  
  *This method has the standard behavior when this object has been created using the standard constructors.*

Public Attributes

- **Token currentToken**
  
  *This is the last token that has been consumed successfully.*

- **int[][] expectedTokenSequences**
  
  *Each entry in this array is an array of integers.*

- **String[] tokenImage**
  
  *This is a reference to the "tokenImage" array of the generated parser within which the parse error occurred.*
Protected Methods

- String **add_escapes** (String str)
  
  Used to convert raw characters to their escaped version when these raw version cannot be used as part of an ASCII string literal.

Protected Attributes

- boolean **specialConstructor**
  
  This variable determines which constructor was used to create this object and thereby affects the semantics of the "getMessage" method (see below).

- String **eol** = System.getProperty("line.separator", "\n")
  
  The end of line string for this machine.

7.11.1 Detailed Description

This exception is thrown when parse errors are encountered.

You can explicitly create objects of this exception type by calling the method generateParseException in the generated parser.

You can modify this class to customize your error reporting mechanisms so long as you retain the public fields.

Definition at line 13 of file ParseException.java.

7.11.2 Constructor & Destructor Documentation

7.11.2.1 io.ParseException.ParseException (Token currentTokenVal, int expectedTokenSequencesVal[[]], String tokenImageVal[])

This constructor is used by the method "generateParseException" in the generated parser.

Calling this constructor generates a new object of this type with the fields "currentToken", "expectedTokenSequences", and "tokenImage" set. The boolean flag "specialConstructor" is also set to true to indicate that this constructor was used to create this object. This constructor calls its super class with the empty string to force the "toString" method of parent class "Throwable" to print the error message in the form: **ParseException** (p.72): <result of getMessage>

Definition at line 27 of file ParseException.java.

7.11.2.2  io.ParseException.ParseException ()  [inline]
The following constructors are for use by you for whatever purpose you can think of.
Constructing the exception in this manner makes the exception behave in the normal way - i.e., as documented in the class "Throwable". The fields "errorToken", "expectedTokenSequences", and "tokenImage" do not contain relevant information. The JavaCC generated code does not use these constructors.
Definition at line 49 of file ParseException.java.
References io.ParseException.specialConstructor.

7.11.2.3  io.ParseException.ParseException  (String  message)  [inline]
Definition at line 54 of file ParseException.java.
References io.ParseException.specialConstructor.

7.11.3  Member Function Documentation

7.11.3.1  String  io.ParseException.add_escapes  (String  str)  [inline, protected]
Used to convert raw characters to their escaped version when these raw version cannot be used as part of an ASCII string literal.
Definition at line 147 of file ParseException.java.
Referenced by io.ParseException.getMessage().

7.11.3.2  String  io.ParseException.getMessage ()  [inline]
This method has the standard behavior when this object has been created using the standard constructors.
Otherwise, it uses "currentToken" and "expectedTokenSequences" to generate a parse error message and returns it. If this object has been created due to a parse error, and you do not catch it (it gets thrown from the parser), then this method is called during the printing of the final stack trace, and hence the correct error message gets displayed.
Definition at line 97 of file ParseException.java.

7.11.4 Member Data Documentation

7.11.4.1 Token io.ParseException.currentToken
This is the last token that has been consumed successfully.
If this object has been created due to a parse error, the token following this token will (therefore) be the first error token.
Definition at line 71 of file ParseException.java.
Referenced by io.ParseException.ParseException().

7.11.4.2 String io.ParseException.eol = System.getProperty("line.separator", "\n") [protected]
The end of line string for this machine.
Definition at line 140 of file ParseException.java.
Referenced by io.ParseException.getMessage().

7.11.4.3 int [][] io.ParseException.expectedTokenSequences
Each entry in this array is an array of integers.
Each array of integers represents a sequence of tokens (by their ordinal values) that is expected at this point of the parse.
Definition at line 78 of file ParseException.java.
Referenced by io.ParseException.getMessage(), and io.ParseException.ParseException().

7.11.4.4 boolean io.ParseException.specialConstructor [protected]
This variable determines which constructor was used to create this object and thereby affects the semantics of the "getMessage" method (see below).
Definition at line 64 of file ParseException.java.
Referenced by io.ParseException.getMessage(), and io.ParseException.ParseException().
7.11.4.5 String [] io.ParseException.tokenImage

This is a reference to the "tokenImage" array of the generated parser within which the parse error occurred.

This array is defined in the generated ...Constants interface.

Definition at line 85 of file ParseException.java.

Referenced by io.ParseException.getMessage(), and io.ParseException.ParseException().

The documentation for this class was generated from the following file:

- ParseException.java

7.12 io.Parser Class Reference

Class to parse spectre files, and add the parsed objects to a scene.

Inheritance diagram for io.Parser:

Collaboration diagram for io.Parser:

Public Methods

- Parser (java.io.InputStream in, ParserInterface pi) throws java.io.FileNotFoundException, java.io.IOException

Create a new parse object attached to the specified file.

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.12 io.Parser Class Reference

- void `ReInit` (java.io.InputStream in, `Parser Interface` p) throws java.io.FileNotFoundException, java.io.IOException
  
  `Reinit the parser.`

- void `parseCurrentFile` () throws io.ParseException, io.SpectreFileError
  
  `Parse the current file, adding new objects by calling the add() method from the Parser Interface (p. 88) used in the constructor.`

- boolean `spectreFile` ()
  
  `Read the first few characters of a file to determine if it is a spectre file.`

- `Parser` (java.io.InputStream stream)
- `Parser` (java.io.Reader stream)
- `Parser` (ParserTokenManager tm)
- void `ReInit` (ParserTokenManager tm)

**Static Public Methods**

- final SceneGraphNode `node` () throws ParseException
- final void `nodeArgList` (SceneGraphNode new_node) throws ParseException
- final void `transformArgList` (SceneGraphNode new_node) throws ParseException
- final void `getDrawingColor` (ParserInterface p) throws ParseException
- final void `colorArgList` (SceneGraphNode new_node) throws ParseException
- final void `polymeshArgList` (SceneGraphNode new_node) throws ParseException
- void `ReInit` (java.io.InputStream stream)
- void `ReInit` (java.io.Reader stream)
- final `Token getNextToken` ()
- final `Token getToken` (int index)
- final `ParseException generateParseException` ()
- final void `enable_tracing` ()
- final void `disable_tracing` ()

**Public Attributes**

- `Parser Interface` add_cmd
Static Public Attributes

- `ParserTokenManager token_source`
- `SimpleCharStream jj_input_stream`
- `Token token`
- `Token jj_ntk`

Static Private Methods

- `final Token jj_consume_token (int kind) throws ParseException`
- `final int jj_ntk ()`

Static Private Attributes

- `boolean jj_initialized_once = false`
- `int jj_ntk`
- `int jj_gen`
- `final int[] jj_la1 = new int[9]`
- `final int[] jj_la1_0 = {0x41,0x280000,0x280000,0x400000,0x800000,0x400000,0x800000,0x800000,0x800000,0x800000}`
- `java.util.Vector jj_expentries = new java.util.Vector()`
- `int[] jj_expentry`
- `int jj_kind = -1`

7.12.1 Detailed Description

Class to parse spectre files, and add the parsed objects to a scene.

Definition at line 18 of file Parser.java.

7.12.2 Constructor & Destructor Documentation


Create a new parse object attached to the specified file.

Parsed objects are added to the scene by calling the add method of the parser interface. This method can only be called once. To parse another file, call the ReInit() (p.82) method.

Parameters:

- `filename` File to parse. If this file cannot be opened, an exception is thrown.
7.12  io(Parser Class Reference)

pi ParserInterface (p. 88) to use when adding new objects to the scene.

Definition at line 33 of file Parser.java.

7.12.2.2  io.Parser (java.io.InputStream stream) [inline]
Definition at line 393 of file Parser.java.
References io.Parser.jj_gen, io.Parser.jj_initialized_once, io.Parser.jj_Jal, and
io.Parser.jj_ntk.

7.12.2.3  io.Parser (java.io.Reader stream) [inline]
Definition at line 418 of file Parser.java.
References io.Parser.jj_gen, io.Parser.jj_initialized_once, io.Parser.jj_Jal, and
io.Parser.jj_ntk.

7.12.2.4  io.Parser (ParserTokenManager tm) [inline]
Definition at line 443 of file Parser.java.
References io.Parser.jj_gen, io.Parser.jj_initialized_once, io.Parser.jj_Jal, and
io.Parser.jj_ntk.

7.12.3  Member Function Documentation

7.12.3.1  final void io.Parser,colorArgList (SceneGraphNode new_-node) throws ParseException [inline, static]
Definition at line 250 of file Parser.java.
Referenced by io.Parser.nodeArgList().

7.12.3.2  final void io.Parser,disable_tracing () [inline, static]
Definition at line 544 of file Parser.java.

7.12.3.3  final void io.Parser,enable_tracing () [inline, static]
Definition at line 541 of file Parser.java.

7.12.3.4  final ParseException io.Parser,generateParseException () [inline, static]
7.12  io.Parser Class Reference

Definition at line 508 of file Parser.java.


Referenced by io.Parser.jj_consume_token().

7.12.3.5  final  void  io.Parser.getDrawingColor (ParserInterface pi) throws ParseException  [inline, static]

Definition at line 236 of file Parser.java.


Referenced by io.Parser.parseCurrentFile().

7.12.3.6  final Token io.Parser.getNextToken ()  [inline, static]

Definition at line 480 of file Parser.java.


Referenced by io.Parser.spectreFile().

7.12.3.7  final Token io.Parser.getToken (int index)  [inline, static]

Definition at line 488 of file Parser.java.

References io.ParserTokenManager.getNextToken(), and io.Token.next.

7.12.3.8  final Token io.Parser.jj_consume_token (int kind) throws ParseException  [inline, static, private]

Definition at line 466 of file Parser.java.


7.12.3.9  final int io.Parser.jj_ntk ()  [inline, static, private]

Definition at line 497 of file Parser.java.

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen


7.12.3.10 final SceneGraphNode io.Parser.node () throws ParseException [inline, static]

Definition at line 122 of file Parser.java.


Referenced by io.Parser.parseCurrentFile().

7.12.3.11 final void io.Parser.nodeArgList (SceneGraphNode new_node) throws ParseException [inline, static]

Definition at line 144 of file Parser.java.


Referenced by io.Parser.node().


Parse the current file, adding new objects by calling the add() method from the Parser Interface (p. 88) used in the constructor.

A variety of exceptions can be thrown by syntax errors in the file.

Definition at line 55 of file Parser.java.


7.12.3.13 final void io.Parser.polymeshArgList (SceneGraphNode new_node) throws ParseException [inline, static]

Definition at line 264 of file Parser.java.

Referenced by io.Parser.nodeArgList().

7.12.3.14  void io.Parser.ReInit (ParserTokenManager tm)  [inline]
Definition at line 458 of file Parser.java.

7.12.3.15  void io.Parser.ReInit (java.io.Reader stream)  [inline, static]
Definition at line 434 of file Parser.java.
Manager.ReInit(), and io.SimpleCharStream.ReInit().

7.12.3.16  void io.Parser.ReInit (java.io.InputStream stream)  
[inline, static]
Definition at line 409 of file Parser.java.
Manager.ReInit(), and io.SimpleCharStream.ReInit().

7.12.3.17  void io.Parser.ReInit (java.io.InputStream in, 
ParserInterface pi)  throws java.io.FileNotFoundException, 
java.io.IOException  [inline]
Reinit the parser.
Equivalent in functionality to the constructor.
Definition at line 43 of file Parser.java.

7.12.3.18  boolean io.Parser.spectreFile ()  [inline]
Read the first few characters of a file to determine if it is a spectre file.
The version of SPECTRE that created the file is also computed, but currently 
this value is not used.

Returns :
    True if this file was created by SPECTRE.
Definition at line 92 of file Parser.java.

### 7.12.3.19 final void io.Parser.transformArgList (SceneGraphNode new_node) throws ParseException
[inline, static]
Definition at line 181 of file Parser.java.

Referenced by `io.Parser.nodeArgList()`.

### 7.12.4 Member Data Documentation

#### 7.12.4.1 Parser Interface io.Parser.add_cmd
Definition at line 19 of file Parser.java.

```java
7.12.4.2 java.util.Vector io.Parser.jj_expentries = new java.util.Vector() [static, private]
```
Definition at line 504 of file Parser.java.
Referenced by `io.Parser.generateParseException()`.

```java
7.12.4.3 int [] io.Parser.jj_expentry [static, private]
```
Definition at line 505 of file Parser.java.
Referenced by `io.Parser.generateParseException()`.

```java
7.12.4.4 int io.Parser.jj_gen [static, private]
```
Definition at line 389 of file Parser.java.

```java
7.12.4.5 boolean io.Parser.jj_initialized_once = false [static, private]
```
Definition at line 384 of file Parser.java.

---

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.12.4.6 `SimpleCharStream` `io.Parser.jj_input_stream` [static]
Definition at line 386 of file Parser.java.

7.12.4.7 `int` `io.Parser.jj_kind = -1` [static, private]
Definition at line 506 of file Parser.java.

7.12.4.8 `final int []` `io.Parser.jj_la1 = new int[9]` [static, private]
Definition at line 390 of file Parser.java.

7.12.4.9 `final int []` `io.Parser.jj_la1_0 = {0x41,0x280000,0x280000,0x400000,0x800000,0x400000,0x800000,0x400000,0x800000}` [static, private]
Definition at line 391 of file Parser.java.
Referenced by `io.Parser.generateParseException()`.

7.12.4.10 `Token` `io.Parser.jj_nt` [static]
Definition at line 387 of file Parser.java.

7.12.4.11 `int` `io.Parser.jj_ntk` [static, private]
Definition at line 388 of file Parser.java.

7.12.4.12 `Token` `io.Parser.token` [static]
Definition at line 387 of file Parser.java.
7.12.4.13  ParserTokenManager io.Parser.token_source  [static]
Definition at line 385 of file Parser.java.
The documentation for this class was generated from the following file:

- Parser.java

7.13  io.ParserConstants Interface Reference

Inheritance diagram for io.ParserConstants:

Public Attributes

- int EOF = 0
- int SPECTRE = 5
- int NODE = 6
- int TRANSFORM = 7
- int COLOR = 8
- int POLYMESH = 9
- int VERTLIST = 10
- int FACELIST = 11
- int EDGELIST = 12
- int CONSTANT = 13
- int FLOAT = 14
- int SCI = 15
- int INTEGER = 16
- int DIGIT = 17
- int SIGN = 18
- int DEFAULT = 0
- String[] tokenImage

7.13.1  Member Data Documentation

7.13.1.1  int io.ParserConstants.COLOR = 8
Definition at line 10 of file ParserConstants.java.
Referenced by io.Parser.nodeArgList().
7.13.1.2 int io.ParserConstants.CONSTANT = 13
Definition at line 15 of file ParserConstants.java.
Referenced by io.Parser.colorArgList(), io.Parser.getDrawingColor(),
io.Parser.polymeshArgList(), io.Parser.spectrumFile(), and io.Parser.transform-
ArgList().

7.13.1.3 int io.ParserConstants.DEFAULT = 0
Definition at line 22 of file ParserConstants.java.

7.13.1.4 int io.ParserConstants.DIGIT = 17
Definition at line 19 of file ParserConstants.java.

7.13.1.5 int io.ParserConstants.EDGELIST = 12
Definition at line 14 of file ParserConstants.java.
Referenced by io.Parser.polymeshArgList().

7.13.1.6 int io.ParserConstants.EOF = 0
Definition at line 6 of file ParserConstants.java.

7.13.1.7 int io.ParserConstants.FACELIST = 11
Definition at line 13 of file ParserConstants.java.
Referenced by io.Parser.polymeshArgList().

7.13.1.8 int io.ParserConstants.FLOAT = 14
Definition at line 16 of file ParserConstants.java.

7.13.1.9 int io.ParserConstants.INTEGER = 16
Definition at line 18 of file ParserConstants.java.

7.13.1.10 int io.ParserConstants.NODE = 6
Definition at line 8 of file ParserConstants.java.
Referenced by io.Parser.node().
7.13.1.11 int io.ParserConstants. POLYMESH = 9
Definition at line 11 of file ParserConstants.java.
Referenced by io.Parser.nodeArgList().

7.13.1.12 int io.ParserConstants.SCI = 15
Definition at line 17 of file ParserConstants.java.

7.13.1.13 int io.ParserConstants.SIGN = 18
Definition at line 20 of file ParserConstants.java.

7.13.1.14 int io.ParserConstants.SPECTRE = 5
Definition at line 7 of file ParserConstants.java.
Referenced by io.Parser.spectreFile().

7.13.1.15 String [] io.ParserConstants.tokenImage
Initial value:

```java
{
    "<EOF>",
    "\n \n",
    "\n\\\n",
    "\\\\n",
    "\\\\\\n",
    "\\\\\\\\n",
    "\\Node:\"",
    "\\Transform:\"
    "\\Color:\",
    "\\Polymesh:\",
    "\\vertexList:\",
    "\\faceList:\",
    "\\edgeList:\",
    "<CONSTANT>",
    "<FLOAT>",
    "<SCI>",
    "<INTEGER>",
    "<DIGIT>",
    "<\n",
    "\\\n",
    "\"null\"
    "\", "\n",
    "\n\n",
    "\n\n",
    "\n\n",
}
```

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
Definition at line 24 of file ParserConstants.java.
Referenced by io.Parser.generateParseException().

7.13.1.16 int io.ParserConstants.TRANSFORM = 7
Definition at line 9 of file ParserConstants.java.
Referenced by io.Parser.nodeArgList().

7.13.1.17 int io.ParserConstants.VERLIST = 10
Definition at line 12 of file ParserConstants.java.
Referenced by io.Parser.polymeshArgList().
The documentation for this interface was generated from the following file:

- ParserConstants.java

7.14 io.ParserInterface Interface Reference

Interface that allows the parser to add objects to the current scene.
Inheritance diagram for io.ParserInterface:

```
io.ParserInterface
      |      |
      v      v
drop.Scene
```

Public Methods

- void add (SceneGraphNode object)
  
  *Add the shape defined by object to the current scene graph.*

- void setColor (Color c)

7.14.1 Detailed Description

Interface that allows the parser to add objects to the current scene.
Definition at line 41 of file ParserInterface.java.
7.14.2 Member Function Documentation

7.14.2.1 void io.ParserInterface.add(SceneGraphNode object)  
Add the shape defined by object to the current scene graph.

Parameters:  
object SceneGraphNode to add to current SceneGraph.

Implemented in disp.Scene (p.131).  
Referenced by io.Parser.parseCurrentFile().

7.14.2.2 void io.ParserInterface setColor (Color c)  
Implemented in disp.Scene (p.136).  
The documentation for this interface was generated from the following file:

- ParserInterface.java

7.15 io.ParserTokenManager Class Reference

Inheritance diagram for io.ParserTokenManager:

Collaboration diagram for io.ParserTokenManager:

Public Methods

- ParserTokenManager (SimpleCharStream stream)
- ParserTokenManager (SimpleCharStream stream, int lexState)

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
Static Public Methods

- void setDebugStream (java.io.PrintStream ds)
- void ReInit (SimpleCharStream stream)
- void ReInit (SimpleCharStream stream, int lexState)
- void SwitchTo (int lexState)
- final Token getNextToken ()

Static Public Attributes

- java.io.PrintStream debugStream = System.out
- final int[] jjnextStates
- final String[] jjstrLiteralImages
- final String[] lexStateNames
- final long[] jjtoToken
- final long[] jjtoSkip
- int curLexState = 0
- int defaultLexState = 0
- int jjnewStateCnt
- int jjround
- int jjmatchedPos
- int jjmatchedKind

Static Protected Attributes

- char curChar

Static Private Methods

- final int jjStopStringLiteralDfa_0 (int pos, long active0)
- final int jjStartNfa_0 (int pos, long active0)
- final int jjStopAtPos (int pos, int kind)
- final int jjStartNfaWithStates_0 (int pos, int kind, int state)
- final int jjMoveStringLiteralDfa0_0 ()
- final int jjMoveStringLiteralDfa1_0 (long active0)
- final int jjMoveStringLiteralDfa2_0 (long old0, long active0)
- final int jjMoveStringLiteralDfa3_0 (long old0, long active0)
- final int jjMoveStringLiteralDfa4_0 (long old0, long active0)
- final int jjMoveStringLiteralDfa5_0 (long old0, long active0)
- final int jjMoveStringLiteralDfa6_0 (long old0, long active0)
- final int jjMoveStringLiteralDfa7_0 (long old0, long active0)
- final int jjMoveStringLiteralDfa8_0 (long old0, long active0)
7.15  io.ParserTokenManager Class Reference

- final int jiMoveStringLiteralDfa9_0 (long old0, long active0)
- final int jiMoveStringLiteralDfa10_0 (long old0, long active0)
- final void jiCheckNAdd (int state)
- final void jiAddStates (int start, int end)
- final void jiCheckNAddTwoStates (int state1, int state2)
- final void jiCheckNAddStates (int start, int end)
- final void jiCheckNAddStates (int start)
- final int jiMoveNfa_0 (int startState, int curPos)
- final void ReInitRounds ()
- final Token jjFillToken ()

Static Private Attributes

- SimpleCharStream input_stream
- final int[] jirounds = new int[18]
- final int[] jistateSet = new int[36]

7.15.1 Constructor & Destructor Documentation

7.15.1.1  io.ParserTokenManager.ParserTokenManager  (SimpleCharStream stream)  [inline]
Definition at line 560 of file ParserTokenManager.java.

7.15.1.2  io.ParserTokenManager.ParserTokenManager  (SimpleCharStream stream, int lexState)  [inline]
Definition at line 566 of file ParserTokenManager.java.
References io.ParserTokenManager.SwitchTo().

7.15.2 Member Function Documentation

7.15.2.1  final Token io.ParserTokenManager.getNextToken ()  [inline, static]
Definition at line 618 of file ParserTokenManager.java.

7.15.2.2 final void io.ParserTokenManager.jjAddStates (int start, int end) [inline, static, private]
Definition at line 338 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.jjMoveNfa_0().

7.15.2.3 final void io.ParserTokenManager.jjCheckNAdd (int state) [inline, static, private]
Definition at line 350 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.jjCheckNAddStates(), io.ParserTokenManager.jjCheckNAddTwoStates(), and io(ParserTokenManager.jjMoveNfa_0().

7.15.2.4 final void io.ParserTokenManager.jjCheckNAddStates (int start) [inline, static, private]
Definition at line 375 of file ParserTokenManager.java.
References io.ParserTokenManager.jjCheckNAdd(), and io(ParserTokenManager.jjnextStates.

7.15.2.5 final void io.ParserTokenManager.jjCheckNAddStates (int start, int end) [inline, static, private]
Definition at line 369 of file ParserTokenManager.java.
References io.ParserTokenManager.jjCheckNAdd(), and io(ParserTokenManager.jjnextStates.
Referenced by io.ParserTokenManager.jjMoveNfa_0().

7.15.2.6 final void io.ParserTokenManager.jjCheckNAddTwoStates (int state1, int state2) [inline, static, private]
Definition at line 364 of file ParserTokenManager.java.
References io.ParserTokenManager.jjCheckNAdd().
Referenced by io.ParserTokenManager.jjMoveNfa0().

7.15.2.7 final Token io.ParserTokenManager.jjFillToken () [inline, static, private]
Definition at line 598 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.getNextToken().

7.15.2.8 final int io.ParserTokenManager.jjMoveNfa_0 (int startState, int curPos) [inline, static, private]
Definition at line 380 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.jjMoveStringLiteralDfa0_0(), io.ParserTokenManager.jjStartNfa_0(), and io.ParserTokenManager.jjStartNfaWithStates_0().

7.15.2.9 final int io.ParserTokenManager.jjMoveStringLiteralDfa0_0 () [inline, static, private]
Definition at line 42 of file ParserTokenManager.java.
References io.ParserTokenManager.jjMoveNfa_0(), io.ParserTokenManager.jjMoveStringLiteralDfa1_0(), and io.ParserTokenManager.jjStopAtPos().
Referenced by io.ParserTokenManager.getNextToken().

7.15.2.10 final int io.ParserTokenManager.jjMoveStringLiteralDfa10_0 (long old0, long active0) [inline, static, private]
Definition at line 330 of file ParserTokenManager.java.
References  io.ParserTokenManager.curChar,  io.ParserTokenManager.inputStream,  io.ParserTokenManager.jjStartNfa_0(),  io.ParserTokenManager.jjStopAtPos(),  io.ParserTokenManager.jjStopStringLiteralDfa_0(), and  io.SimpleCharStream.readChar().

Referenced by  io.ParserTokenManager.jjMoveStringLiteralDfa9_0().

### 7.15.2.11 final int  io.ParserTokenManager.jjMoveStringLiteralDfa1_0 (long  \textit{active0})  [inline, static, private]

Definition at line 78 of file ParserTokenManager.java.

References  io.ParserTokenManager.curChar,  io.ParserTokenManager.inputStream,  io.ParserTokenManager.jjMoveStringLiteralDfa2_0(),  io.ParserTokenManager.jjStartNfa_0(),  io.ParserTokenManager.jjStopStringLiteralDfa_0(), and  io.SimpleCharStream.readChar().

Referenced by  io.ParserTokenManager.jjMoveStringLiteralDfa0_0().

### 7.15.2.12 final int  io.ParserTokenManager.jjMoveStringLiteralDfa2_0 (long  \textit{old0}, long  \textit{active0})  [inline, static, private]

Definition at line 106 of file ParserTokenManager.java.

References  io.ParserTokenManager.curChar,  io.ParserTokenManager.inputStream,  io.ParserTokenManager.jjMoveStringLiteralDfa3_0(),  io.ParserTokenManager.jjStartNfa_0(),  io.ParserTokenManager.jjStopStringLiteralDfa_0(), and  io.SimpleCharStream.readChar().

Referenced by  io.ParserTokenManager.jjMoveStringLiteralDfa1_0().

### 7.15.2.13 final int  io.ParserTokenManager.jjMoveStringLiteralDfa3_0 (long  \textit{old0}, long  \textit{active0})  [inline, static, private]

Definition at line 136 of file ParserTokenManager.java.

References  io.ParserTokenManager.curChar,  io.ParserTokenManager.inputStream,  io.ParserTokenManager.jjMoveStringLiteralDfa4_0(),  io.ParserTokenManager.jjStartNfa_0(),  io.ParserTokenManager.jjStopAtPos(),  io.ParserTokenManager.jjStopStringLiteralDfa_0(), and  io.SimpleCharStream.readChar().

Referenced by  io.ParserTokenManager.jjMoveStringLiteralDfa2_0().

### 7.15.2.14 final int  io.ParserTokenManager.jjMoveStringLiteralDfa4_0 (long  \textit{old0}, long  \textit{active0})  [inline, static, private]

Definition at line 168 of file ParserTokenManager.java.
References io.ParserTokenManager.curChar, io.ParserTokenManager.inputStream, io.ParserTokenManager.jjMoveStringLiteralDfa5_0, io.ParserTokenManager.jjStartNfa_0, io.ParserTokenManager.jjStopAtPos(), io.ParserTokenManager.jjStopStringLiteralDfa_0(), and io.SimpleCharStream.readChar().

Referenced by io.ParserTokenManager.jjMoveStringLiteralDfa3_0().

7.15.2.15 final int io.ParserTokenManager.jjMoveStringLiteralDfa5_0 (long old0, long active0) [inline, static, private]

Definition at line 200 of file ParserTokenManager.java.

References io.ParserTokenManager.curChar, io.ParserTokenManager.inputStream, io.ParserTokenManager.jjMoveStringLiteralDfa6_0, io.ParserTokenManager.jjStartNfa_0, io.ParserTokenManager.jjStopAtPos(), io.ParserTokenManager.jjStopStringLiteralDfa_0(), and io.SimpleCharStream.readChar().

Referenced by io.ParserTokenManager.jjMoveStringLiteralDfa4_0().

7.15.2.16 final int io.ParserTokenManager.jjMoveStringLiteralDfa6_0 (long old0, long active0) [inline, static, private]

Definition at line 230 of file ParserTokenManager.java.

References io.ParserTokenManager.curChar, io.ParserTokenManager.inputStream, io.ParserTokenManager.jjMoveStringLiteralDfa7_0, io.ParserTokenManager.jjStartNfa_0, io.ParserTokenManager.jjStopAtPos(), io.ParserTokenManager.jjStopStringLiteralDfa_0(), and io.SimpleCharStream.readChar().

Referenced by io.ParserTokenManager.jjMoveStringLiteralDfa5_0().

7.15.2.17 final int io.ParserTokenManager.jjMoveStringLiteralDfa7_0 (long old0, long active0) [inline, static, private]

Definition at line 256 of file ParserTokenManager.java.

References io.ParserTokenManager.curChar, io.ParserTokenManager.inputStream, io.ParserTokenManager.jjMoveStringLiteralDfa8_0, io.ParserTokenManager.jjStartNfa_0, io.ParserTokenManager.jjStopStringLiteralDfa_0(), and io.SimpleCharStream.readChar().

Referenced by io.ParserTokenManager.jjMoveStringLiteralDfa6_0().

7.15.2.18 final int io.ParserTokenManager.jjMoveStringLiteralDfa8_0 (long old0, long active0) [inline, static, private]
Definition at line 280 of file ParserTokenManager.java.
References io.ParserTokenManager.curChar, io.ParserTokenManager.inputStream, io.ParserTokenManager.jjMoveStringLiteralDfa9_0(), io.ParserTokenManager.jjStartNfa_0(), io.ParserTokenManager.jjStopAtPos(), io.ParserTokenManager.jjStopStringLiteralDfa_0(), and io.SimpleCharStream.readChar().

Referenced by io.ParserTokenManager.jjMoveStringLiteralDfa7_0().

7.15.2.19 final int io.ParserTokenManager.jjMoveStringLiteralDfa9_0 (long old0, long active0) [inline, static, private]
Definition at line 308 of file ParserTokenManager.java.
References io.ParserTokenManager.curChar, io.ParserTokenManager.inputStream, io.ParserTokenManager.jjMoveStringLiteralDfa10_0(), io.ParserTokenManager.jjStartNfa_0(), io.ParserTokenManager.jjStopAtPos(), io.ParserTokenManager.jjStopStringLiteralDfa_0(), and io.SimpleCharStream.readChar().

Referenced by io.ParserTokenManager.jjMoveStringLiteralDfa8_0().

7.15.2.20 final int io.ParserTokenManager.jjStartNfa_0 (int pos, long active0) [inline, static, private]
Definition at line 24 of file ParserTokenManager.java.
References io.ParserTokenManager.jjMoveNfa_0(), and io.ParserTokenManager.jjStopStringLiteralDfa_0().


7.15.2.21 final int io.ParserTokenManager.jjStartNfaWithStates_0 (int pos, int kind, int state) [inline, static, private]
Definition at line 34 of file ParserTokenManager.java.

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.15.2.22 final int io.PARSER_TOKEN_MANAGER\_jjStopAtPos (int pos, int kind) [inline, static, private]
Definition at line 28 of file ParserTokenManager.java.
References io.ParserTokenManager[jjmatchedKind], and io.ParserTokenManager[jjmatchedPos].
Referenced by io.ParserTokenManager[jjMoveStringLiteralDfa0_0()], io.ParserTokenManager[jjMoveStringLiteralDfa10_0()], io.ParserTokenManager[jjMoveStringLiteralDfa11_0()], io.ParserTokenManager[jjMoveStringLiteralDfa12_0], io.ParserTokenManager[jjMoveStringLiteralDfa13_0], io.ParserTokenManager[jjMoveStringLiteralDfa14_0], io.ParserTokenManager[jjMoveStringLiteralDfa15_0], io.ParserTokenManager[jjMoveStringLiteralDfa16_0], io.ParserTokenManager[jjMoveStringLiteralDfa17_0], io.ParserTokenManager[jjMoveStringLiteralDfa18_0], and io.ParserTokenManager[jjStartNfa_0].

7.15.2.23 final int io.PARSER_TOKEN_MANAGER\_jjStopStringLiteralDfa_0 (int pos, long active0) [inline, static, private]
Definition at line 16 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager[jjMoveStringLiteralDfa10_0], io.ParserTokenManager[jjMoveStringLiteralDfa11_0], io.ParserTokenManager[jjMoveStringLiteralDfa12_0], io.ParserTokenManager[jjMoveStringLiteralDfa13_0], io.ParserTokenManager[jjMoveStringLiteralDfa14_0], io.ParserTokenManager[jjMoveStringLiteralDfa15_0], io.ParserTokenManager[jjMoveStringLiteralDfa16_0], io.ParserTokenManager[jjMoveStringLiteralDfa17_0], io.ParserTokenManager[jjMoveStringLiteralDfa18_0], and io.ParserTokenManager[jjStartNfa_0].

7.15.2.24 void io.ParserTokenManager.ReInit (SimpleCharStream stream, int lexState) [inline, static]
Definition at line 585 of file ParserTokenManager.java.
References io.ParserTokenManager.ReInit(), and io.ParserTokenManager.SwitchTo().

7.15.2.25 void io.ParserTokenManager.ReInit (SimpleCharStream stream) [inline, static]
Definition at line 571 of file ParserTokenManager.java.
References io.ParserTokenManager.curLexState, io.ParserTokenManager.defaultLexState, io.ParserTokenManager[jjmatchedPos], io.ParserTokenManager[jjnewStateCnt], and io.ParserTokenManager.ReInitRounds().
Referenced by io.ParserTokenManager.ReInit(), and io.Parser.ReInit().

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.15.2.26 final void io(ParserTokenManager.ReInitRounds) [inline, static, private]
Definition at line 578 of file ParserTokenManager.java.
References io.ParserTokenManager.jjround, and io.ParserTokenManager.jjrounds.
Referenced by io.ParserTokenManager.jjMoveNfa(0), and io.ParserTokenManager.ReInit().

7.15.2.27 void io(ParserTokenManager.setDebugStream(java.io.PrintStream ds) [inline, static]
Definition at line 15 of file ParserTokenManager.java.
References io.ParserTokenManager.debugStream.

7.15.2.28 void io(ParserTokenManager.SwitchTo(int lexState) [inline, static]
Definition at line 590 of file ParserTokenManager.java.
References io.ParserTokenManager.curLexState.
Referenced by io.ParserTokenManager.ParserTokenManager(), and io.ParserTokenManager.ReInit().

7.15.3 Member Data Documentation

7.15.3.1 char io(ParserTokenManager.curChar [static, protected]
Definition at line 559 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.getNextToken(), io.ParserTokenManager.jjMoveNfa(0), io.ParserTokenManager.jjMoveStringLiteralDfa10(0), io.ParserTokenManager.jjMoveStringLiteralDfa11(0), io.ParserTokenManager.jjMoveStringLiteralDfa2(0), io.ParserTokenManager.jjMoveStringLiteralDfa3(0), io.ParserTokenManager.jjMoveStringLiteralDfa4(0), io.ParserTokenManager.jjMoveStringLiteralDfa5(0), io.ParserTokenManager.jjMoveStringLiteralDfa6(0), io.ParserTokenManager.jjMoveStringLiteralDfa7(0), io.ParserTokenManager.jjMoveStringLiteralDfa8(0), io.ParserTokenManager.jjMoveStringLiteralDfa9(0), and io.ParserTokenManager.jjStartNfaWithStates(0).

7.15.3.2 int io(ParserTokenManager.curLexState = 0 [static]
Definition at line 611 of file ParserTokenManager.java.
Referenced by `io.ParserTokenManager.nextToken()`, `io.ParserTokenManager.reInit()`, and `io.ParserTokenManager.switchTo()`.

### 7.15.3.3 `java.io.PrintStream io.ParserTokenManager.debugStream = System.out` [static]

Definition at line 14 of file `ParserTokenManager.java`.

Referenced by `io.ParserTokenManager.setDebugStream()`.

### 7.15.3.4 `int io.ParserTokenManager.defaultLexState = 0` [static]

Definition at line 612 of file `ParserTokenManager.java`.

Referenced by `io.ParserTokenManager.reInit()`.

### 7.15.3.5 `SimpleCharStream io.ParserTokenManager.inputStream` [static, private]

Definition at line 556 of file `ParserTokenManager.java`.


### 7.15.3.6 `int io.ParserTokenManager.jjmatchedKind` [static]

Definition at line 616 of file `ParserTokenManager.java`.

Referenced by `io(ParserTokenManager.nextToken()`, `io(ParserTokenManager.jjFillToken()`, `io(ParserTokenManager.jjMoveNfa0()`, `io.ParserTokenManager.jjStartNfaWithStates_0()`, and `io.ParserTokenManager.jjStopAtPos()`.

### 7.15.3.7 `int io.ParserTokenManager.jjmatchedPos` [static]

Definition at line 615 of file `ParserTokenManager.java`.

Referenced by `io(ParserTokenManager.nextToken()`, `io(ParserTokenManager.jjMoveNfa0()`, `io(ParserTokenManager.jjStartNfaWithStates_0()`, `io.ParserTokenManager.jjStopAtPos()`, and `io.ParserTokenManager.reInit()`.
7.15.3.8 int io.ParserTokenManager.jjnewStateCnt [static]
Definition at line 613 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.jjAddStates(), io.ParserTokenManager.jjCheckNAdd(), io.ParserTokenManager.jjMoveNfa,0(), and io.ParserTokenManager.ReInit().

7.15.3.9 final int [] io.ParserTokenManager.jjnextStates [static]
Initial value:

    { 
        7, 8, 9, 13, 14, 16, 8, 11, 13, 12, 16, 8, 9, 2, 
        3, 4, 
    }

Definition at line 537 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.jjAddStates(), and io.ParserTokenManager.jjCheckNAddStates().

7.15.3.10 int io.ParserTokenManager.jjround [static]
Definition at line 614 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.jjCheckNAdd(), io.ParserTokenManager.jjMoveNfa,0(), and io.ParserTokenManager.ReInitRounds().

7.15.3.11 final int [] io.ParserTokenManager.jjrounds = new int[18]
[static, private]
Definition at line 557 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.jjCheckNAdd(), and io.ParserTokenManager.ReInitRounds().

7.15.3.12 final int [] io.ParserTokenManager.jjstateSet = new int[36]
[static, private]
Definition at line 558 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.jjAddStates(), io.ParserTokenManager.jjCheckNAdd(), and io.ParserTokenManager.jjMoveNfa,0().

7.15.3.13 final String [] io.ParserTokenManager.jjstrLiteralImages
[static]
Initial value:
7.15  io.ParserTokenManager Class Reference

```java
{
"", null, null, null, null, "\123\120\105\103\124\122\105",
"\116\157\144\145\172", "\124\122\141\156\163\146\157\162\155\172", "\103\157\154\157\152\172",
"\120\157\151\155\146\163\150\172", "\166\145\162\164\145\170\114\151\163\164\172",
"\140\141\143\151\163\164\172", "\145\144\147\145\151\163\164\172", null, null, null, null, null,
"\173", "\175", "\156\156\154\154", "\54", "\55", "\51", }
```

Definition at line 541 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.jjFillToken().

7.15.3.14  final long [] io.ParserTokenManager.ijtToSkip  [static]
Initial value:

```java
{
  oxieL,
}
```

Definition at line 553 of file ParserTokenManager.java.

7.15.3.15  final long [] io.ParserTokenManager.ijtToken  [static]
Initial value:

```java
{
  oxfSfeIL,
}
```

Definition at line 550 of file ParserTokenManager.java.
Referenced by io.ParserTokenManager.getNextToken().

7.15.3.16  final String [] io.ParserTokenManager.lexStateNames  [static]
Initial value:

```java
{
  "DEFAULT",
}
```

Definition at line 547 of file ParserTokenManager.java.
The documentation for this class was generated from the following file:

- ParserTokenManager.java

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.16 disp.PickablePolygon Class Reference

Creates a drawable polygon with a reference back to the SceneGraph.

Public Methods

- **PickablePolygon** (Polygon p, SceneGraphNode sgn, Color c)
  
  Makes a new pickable polygon for use with mouse selection.

- **PickablePolygon** (Polymesh.Vertex[] verts, SceneGraphNode sgn, Color c)
  
  Returns a pickable polygon created from connecting the vertices in the verts array in counterclockwise order.

- String **toString** ()
  
  Creates a string of all the points of the current polygon.

- Polygon **getPolygon** ()
  
  Returns the current pickable polygon.

- SceneGraphNode **getSceneGraphNode** ()
  
  Returns the reference.

- Color **getColor** ()
  
  Returns the current color of the polygon.

Public Attributes

- Polygon **polygon**
  
  Polygon to draw in slice window.

- SceneGraphNode **pickReference**
  
  Reference to node in the scene graph.

- Color **color**
  
  Color of the polygon.

7.16.1 Detailed Description

Creates a drawable polygon with a reference back to the SceneGraph.

Definition at line 45 of file PickablePolygon.java.
7.16.2 Constructor & Destructor Documentation

7.16.2.1 disp.PickablePolygon.PickablePolygon (Polygon p, SceneGraphNode sgn, Color c) [inline]

Makes a new pickable polygon for use with mouse selection.

Parameters:
  - p The picked polygon
  - sgn Reference for the polygon back into the scene graph
  - c Color of the polygon

Definition at line 56 of file PickablePolygon.java.

7.16.2.2 disp.PickablePolygon.PickablePolygon (Polymesh,Vertex verts[], SceneGraphNode sgn, Color c) [inline]

Returns a pickable polygon created from connecting the vertices in the verts array in counterclockwise order.

Parameters:
  - verts Array of vertexes of the polygon
  - sgn Reference for polygon back into the SceneGraph
  - c Color of the polygon

Definition at line 69 of file PickablePolygon.java.

7.16.3 Member Function Documentation

7.16.3.1 Color disp.PickablePolygon.getColor () [inline]

Returns the current color of the polygon.

Returns:
  - Current color

Definition at line 116 of file PickablePolygon.java.
References disp.PickablePolygon.color.
7.16.3.2 Polygon disp.PickablePolygon.getPolygon () [inline]
Returns the current pickable polygon.

Returns:
The current polygon

Definition at line 100 of file PickablePolygon.java.
References disp.PickablePolygon.polygon.
Referenced by disp.Scene.selectObject().

7.16.3.3 SceneGraphNode disp.PickablePolygon.getSceneGraphNode () [inline]
Returns the reference.

Returns:
Returns the reference for the polygon back into the scene graph

Definition at line 108 of file PickablePolygon.java.
References disp.PickablePolygon.pickReference.
Referenced by disp.Scene.selectObject().

7.16.3.4 String disp.PickablePolygon.toString () [inline]
Creates a string of all the points of the current polygon.

Returns:
String consisting of all the points in the polygon

Definition at line 88 of file PickablePolygon.java.
References disp.PickablePolygon.polygon.

7.16.4 Member Data Documentation

7.16.4.1 Color disp.PickablePolygon.color
Color of the polygon.
Definition at line 48 of file PickablePolygon.java.
Referenced by disp.PickablePolygon.getColor(), and disp.PickablePolygon.PickablePolygon().
7.16.4.2 SceneGraphNode disp.PickablePolygon.pickReference
Reference to node in the scene graph.
Definition at line 47 of file PickablePolygon.java.
Referenced by disp.PickablePolygon.getSceneGraphNode(), and disp.PickablePolygon.PickablePolygon().

7.16.4.3 Polygon disp.PickablePolygon.polygon
Polygon to draw in slice window.
Definition at line 46 of file PickablePolygon.java.
Referenced by disp.PickablePolygon.getPolygon(), disp.PickablePolygon.PickablePolygon(), and disp.PickablePolygon.toString().
The documentation for this class was generated from the following file:

  • PickablePolygon.java

7.17 win.PickingMouseListener Class Reference

Handles events thrown by user input via mouse.

Inheritance diagram for win.PickingMouseListener:

```
MouseAdapter

win.PickingMouseListener
```

Collaboration diagram for win.PickingMouseListener:

```
MouseAdapter

w...
7.17 win.PickingMouseListener Class Reference

- void mouseClicked (MouseEvent evt)
  
  Handles events for each button.

Private Attributes

- Vector objListReference = null

7.17.1 Detailed Description

Handles events thrown by user input via mouse.
Definition at line 46 of file PickingMouseListener.java.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 win.PickingMouseListener.PickingMouseListener (Vector objRef) [inline]

Create a new PickingMouseListener (p. 105) with the specified polygon reference.

Each mouse listener needs a reference to its ControlWindow (p. 18)'s PickablePolygon list. This constructor is a wrapper around the normal MouseAdapter constructor that stores this reference.

Parameters:

  objRef The Vector whose first element is the ControlWindow (p. 18)'s PickablePolygon array representing the objects currently visible.

Definition at line 60 of file PickingMouseListener.java.
References win.PickingMouseListener.objListReference.

7.17.3 Member Function Documentation

7.17.3.1 void win.PickingMouseListener.mouseClicked (MouseEvent evt) [inline]

Handles events for each button.

Parameters:

  evt The mouse event to handle.

Definition at line 70 of file PickingMouseListener.java.
References win.PickingMouseListener.objListReference.
7.17.4 Member Data Documentation

7.17.4.1 Vector win.PickingMouseListener.objListReference = null [private]

Definition at line 47 of file PickingMouseListener.java.

Referenced by win.PickingMouseListener.mouseClicked(), and win.PickingMouseListener.PickingMouseListener().

The documentation for this class was generated from the following file:

- PickingMouseListener.java

7.18 obj.Polymesh Class Reference

A mesh of triangles enclosing a solid region of space.

Collaboration diagram for obj.Polymesh:

```
\begin{center}
\begin{tikzpicture}
  \node (v1) at (0,0) {obj.Polymesh.Vertex
  \node (v2) at (1,0) {v2 \node (v1) at (0,0) {v1
  \node (v3) at (0,1) {\textbf{obj.Polymesh.Line
  \node (v4) at (1,1) {\textbf{obj.Polymesh.Tri
  \node (v5) at (0,2) {\textbf{obj.Polymesh

\draw [->] (v1) edge (v2);
\draw [->] (v2) edge (v3);
\draw [->] (v3) edge (v4);
\draw [->] (v4) edge (v5);
\draw [->] (v5) edge (v1);
\end{tikzpicture}
\end{center}
```

Public Methods

- **Polymesh** (Vertex[] vertexList, Tri[] faceList, Line[] edgeList)

  Create a new polymesh.

- **toString** ()

  Return a string representing the polymesh.

- **Vertex[] slice** (Plane plane, Transform t)

  Slice the Polymesh (p. 107), after applying the transformation matrix t to all vertices.

Static Public Methods

- **Line[] edgeify** (Vertex[] vList, Tri[] fList)
Construct an edge list for a \texttt{Polymesh} (p. 107), once the vertex list and face list are completed.

Public Attributes

- \texttt{Vertex[]} \texttt{verts} = null
  
  \textit{List of vertices.}

- \texttt{Tri[]} \texttt{fList} = null
  
  \textit{List of (triangular) faces.}

- \texttt{Line[]} \texttt{eList} = null
  
  \textit{List of edges.}

Protected Methods

- double \texttt{dot} (\texttt{Vertex} v1, \texttt{Vertex} v2)
  
  \textit{Return the dot product of the vectors going from the origin to the points \texttt{v1} and \texttt{v2}.}

- \texttt{Vertex} \texttt{vertSub} (\texttt{Vertex} v1, \texttt{Vertex} v2)
  
  \textit{Create a new vertex from \texttt{v1} - \texttt{v2}.}

- \texttt{Vertex} \texttt{vertAdd} (\texttt{Vertex} v1, \texttt{Vertex} v2)
  
  \textit{Create a new vertex from \texttt{v1} + \texttt{v2}.}

- \texttt{Vertex} \texttt{vertMult} (\texttt{Vertex} v1, double t)
  
  \textit{Create a new vertex from \texttt{t} * \texttt{v1}.}

Protected Attributes

- double \texttt{xMin}
  
  \textit{Bounding box for the \texttt{Polymesh} (p. 107).}

- double \texttt{yMin}
  
  \textit{Bounding box for the \texttt{Polymesh} (p. 107).}

- double \texttt{xMax}
  
  \textit{Bounding box for the \texttt{Polymesh} (p. 107).}
• double yMax  
  *Bounding box for the Polymesh (p. 107).*

Private Methods

• Line transformLine (Line l, Transform t)  
  *Apply transform t to line l, and return the resulting line.*

• Vertex[] intersectLinePlane (Line l, Plane plane)  
  *Return either 0, 1, or 2 vertices resulting from intersecting line segment l with the plane.*

7.18.1 Detailed Description

A mesh of triangles enclosing a solid region of space.

A Polymesh (p. 107) can be sliced and transformed.

Definition at line 48 of file Polymesh.java.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 obj.Polymesh.Polymesh (Vertex vertexList[], Tri faceList[], Line edgeList[]) [inline]

Create a new polymesh.

Parameters:
  * vertexList A list of all vertices in the polymesh
  * faceList A list of all triangular faces that compose the polymesh. The vertices should be indices into the vertexList array.
  * edgeList A list of the edges in the polymesh. This array must be obtained from a call to Polymesh.edgeify().

Definition at line 257 of file Polymesh.java.

References obj.Polymesh.Vertex.setIndex().

7.18.3 Member Function Documentation

7.18.3.1 double obj.Polymesh.dot (Vertex v1, Vertex v2) [inline, protected]
Return the dot product of the vectors going from the origin to the points v1 and v2.

**Parameters:**
- v1 First vector
- v2 Second vector

**Returns:**
- Dot product of the two vectors

Definition at line 352 of file Polymesh.java.
Referenced by obj.Polymesh.intersectLinePlane().

### 7.18.3.2 Line [] obj.Polymesh.edgeify (Vertex vList[], Tri fList[]) [inline, static]

Construct an edge list for a Polymesh (p.107), once the vertex list and face list are completed.
This _must_ be done before the Polymesh (p.107) can be sliced.

**Parameters:**
- vList List of vertices
- fList List of faces

**Returns:**
- List of all edges in the polymesh

Definition at line 277 of file Polymesh.java.
References obj.Polymesh.Tri.v.
Referenced by obj.ObjFactory.newCube().

### 7.18.3.3 Vertex [] obj.Polymesh.intersectLinePlane (Line l, Plane plane) [inline, private]

Return either 0, 1, or 2 vertices resulting from intersecting line segment l with the plane.
Unused elements in the array are set to null.

**Parameters:**
- l Line (p.117) to intersect
- plane Plane (p.119) to intersect
Returns:
Either 0, 1, or 2 vertices from the intersection

Definition at line 602 of file Polymesh.java.
Referenced by obj.Polymesh.slice().

7.18.3.4 Vertex [] obj.Polymesh.slice (Plane plane, Transform t) [inline]
Slice the Polymesh (p. 107), after applying the transformation matrix t to all vertices.
Points are returned in counter-clockwise order from the positive x-axis. You must call edgeify() (p. 110) before using this routine.

Parameters:
plane Slice plane
t Transformation matrix to apply to the polymesh before slicing

Returns:
Sorted array of points describing the slice

Definition at line 406 of file Polymesh.java.
Referenced by obj.SceneGraphNode.slice().

7.18.3.5 String obj.Polymesh.toString () [inline]
Return a string representing the polymesh.

Returns:
A string listing all faces on a separate line

Definition at line 319 of file Polymesh.java.
7.18.3.6 Line obj:Polymesh.transformLine (Line l, Transform t)  
[inline, private]  
Apply transform t to line l, and return the resulting line.

Parameters:
  
l Line (p.117) to transform  
t Transform (p.180) to apply

Returns:
  
Transformed line

Definition at line 588 of file Polymesh.java.

References obj:Transform.apply(), obj:Polymesh:Line:v1, and  

Referenced by obj:Polymesh:intersect().

7.18.3.7 Vertex obj:Polymesh.vertAdd (Vertex v1, Vertex v2)  
[inline, protected]  
Create a new vertex from v1 + v2.

Parameters:
  
v1 First vertex  
v2 Second vertex

Returns:
  
v1 + v2

Definition at line 377 of file Polymesh.java.

References obj:Polymesh:Vertex:coords.

Referenced by obj:Polymesh:intersectLinePlane().

7.18.3.8 Vertex obj:Polymesh.vertMult (Vertex v1, double t)  
[inline, protected]  
Create a new vertex from t*v1.

Parameters:
  
v1 Vertex (p.122) to scale  
t Constant by which to scale v1

Returns:
  
t*v1
Definition at line 390 of file Polymesh.java.

7.18.3.9 Vertex obj.Polymesh.vertSub (Vertex v1, Vertex v2) [inline, protected]
Create a new vertex from v1 - v2.

Parameters:
   v1 First vertex
   v2 Second vertex

Returns :
   v1 - v2

Definition at line 364 of file Polymesh.java.
Referenced by obj.Polymesh.intersectLinePlane().

7.18.4 Member Data Documentation

7.18.4.1 Line [] obj.Polymesh.eList = null
List of edges.
This array contains redundant information, but it is needed to accommodate
slicing.
Definition at line 243 of file Polymesh.java.

7.18.4.2 Tri [] obj.Polymesh.fList = null
List of (triangular) faces.
Definition at line 242 of file Polymesh.java.
Referenced by obj.SceneGraph.getGeometryArray().

7.18.4.3 Vertex [] obj.Polymesh.verts = null
List of vertices.
Definition at line 241 of file Polymesh.java.
7.18.4.4 double obj.PolyMesh.xMax [protected]
Bounding box for the PolyMesh (p. 107).
Definition at line 51 of file PolyMesh.java.
Referenced by obj.PolyMesh.slice(), and obj.SceneGraphNode.updateBounds().

7.18.4.5 double obj.PolyMesh.xMin [protected]
Bounding box for the PolyMesh (p. 107).
Definition at line 49 of file PolyMesh.java.
Referenced by obj.PolyMesh.slice(), and obj.SceneGraphNode.updateBounds().

7.18.4.6 double obj.PolyMesh.yMax [protected]
Bounding box for the PolyMesh (p. 107).
Definition at line 52 of file PolyMesh.java.
Referenced by obj.PolyMesh.slice(), and obj.SceneGraphNode.updateBounds().

7.18.4.7 double obj.PolyMesh.yMin [protected]
Bounding box for the PolyMesh (p. 107).
Definition at line 50 of file PolyMesh.java.
Referenced by obj.PolyMesh.slice(), and obj.SceneGraphNode.updateBounds().
The documentation for this class was generated from the following file:

- PolyMesh.java

7.19 obj.PolyMesh.HullSort Class Reference
A comparator that can sort points in counter-clockwise order around a point. Inheritance diagram for obj.PolyMesh.HullSort:
Collaboration diagram for `obj.Polymesh.HullSort`:

![Collaboration diagram](image)

**Public Methods**

- **HullSort** (double x, double y)  
  Create a new HullSort (p. 114) object that sorts about (x,y).

- **int compare** (Object o1, Object o2)  
  The comparison function for the Comparator interface.

**Public Attributes**

- **double xCenter**  
  X cord of point about which to sort.

- **double yCenter**  
  Y cord of point about which to sort.

### 7.19.1 Detailed Description

A comparator that can sort points in counter-clockwise order around a point.

Definition at line 488 of file Polymesh.java.

### 7.19.2 Constructor & Destructor Documentation

#### 7.19.2.1 `obj.Polymesh.HullSort.HullSort (double x, double y)`

Create a new HullSort (p. 114) object that sorts about (x,y).

**Parameters:**

- **x** x-coordinate of sort point
- **y** y-coordinate of sort point
Definition at line 499 of file Polymesh.java.

7.19.3 Member Function Documentation

7.19.3.1 int obj.Polymesh.HullSort.compare (Object o1, Object o2) [inline]
The comparison function for the Comparator interface.

Parameters:
o1 First object to compare (cast to a Vertex (p. 122))
o2 Second object to compare (cast to a Vertex (p. 122))

Returns :
1 if o1 > o2, -1 if o1 < o2, or 0 if o1 == o2

Definition at line 511 of file Polymesh.java.

7.19.4 Member Data Documentation

7.19.4.1 double obj.Polymesh.HullSort.xCenter
X cord of point about which to sort.

Definition at line 489 of file Polymesh.java.

7.19.4.2 double obj.Polymesh.HullSort.yCenter
T cord of point about which to sort.

Definition at line 490 of file Polymesh.java.
The documentation for this class was generated from the following file:

- Polymesh.java
7.20 obj.Polymesh.Line Class Reference

A line segment in 3D space.

Collaboration diagram for obj.Polymesh.Line:

```
+-------------------------+
| obj.Polymesh.Vertex    |
|                         |
| v2                      |
|                         |
| v1                      |
|                         |
+-------------------------+
```

**Public Methods**

- `Line (Vertex vert1, Vertex vert2)`
  
  Create a new line segment connecting two vertices.

- boolean `equals (Object obj)`
  
  Returns true if the two lines have the same vertices at their endpoints.

- int `hashCode ()`
  
  Compute a hash code for this line.

- String `toString ()`
  
  Return a string representing the line.

**Public Attributes**

- `Vertex v1`
  
  Start point of line segment.

- `Vertex v2`
  
  End point of the line segment.

7.20.1 Detailed Description

A line segment in 3D space.

Definition at line 117 of file Polymesh.java.
7.20.2 Constructor & Destructor Documentation

7.20.2.1 obj.Polymesh.Line.Line (Vertex vert1, Vertex vert2) [inline]
Create a new line segment connecting two vertices.

Parameters:
vert1 First vertex
vert2 Second vertex

Definition at line 127 of file Polymesh.java.

7.20.3 Member Function Documentation

7.20.3.1 boolean obj.Polymesh.Line.equals (Object obj) [inline]
Returns true if the two lines have the same vertices at their endpoints.
The vertices are checked for equality by comparing their \textit{references}, not
their fields!

Parameters:
obj Line (p.117) to compare with this line. obj will be immediately to a
Line (p.117), throwing a fatal exception if this is not possible

Returns :
True if obj is the same line as this object

Definition at line 144 of file Polymesh.java.

7.20.3.2 int obj.Polymesh.Line.hashCode () [inline]
Compute a hash code for this line.
Two lines should have the same hash code if they have the same vertices, no
matter which order they are listed in.

Returns :
Hashcode for this line, suitable for use in a HashTable

Definition at line 159 of file Polymesh.java.
7.20.3.3 String obj.Polymesh.Line.toString () [inline]
Return a string representing the line.

Returns:
A string of the form ”vertex.toString() – vertex.toString()”

Definition at line 178 of file Polymesh.java.
References obj.Polymesh.Vertex.getIndex().

7.20.4 Member Data Documentation

7.20.4.1 Vertex obj.Polymesh.Line.v1
Start point of line segment.
Definition at line 118 of file Polymesh.java.
Referenced by obj.Polymesh.Line.equals(), obj.Polymesh.intersectLinePlane(),
obj.Polymesh.slice(), and obj.Polymesh.transformLine().

7.20.4.2 Vertex obj.Polymesh.Line.v2
End point of the line segment.
Definition at line 119 of file Polymesh.java.
Referenced by obj.Polymesh.Line.equals(), obj.Polymesh.intersectLinePlane(),
obj.Polymesh.slice(), and obj.Polymesh.transformLine().

The documentation for this class was generated from the following file:

- Polymesh.java

7.21 obj.Polymesh.Plane Class Reference

A plane in 3D space.
Collaboration diagram for obj.Polymesh.Plane:

```
    obj.Polymesh.Plane
      | normal
      | point
    obj.Polymesh.Vertex
```

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
Public Methods

- **Plane (Vertex norm, Vertex p)**
  
  *Create a new plane passing through the point p, normal to the vector norm.*

Public Attributes

- **Vertex normal = null**
  
  *Normal vector to the plane.*

- **Vertex point = null**
  
  *Point on the plane.*

7.21.1 Detailed Description

A plane in 3D space.

Definition at line 224 of file Polymesh.java.

7.21.2 Constructor & Destructor Documentation


[in-line]

Create a new plane passing through the point p, normal to the vector norm.

**Parameters:**

- **norm** Normal vector to the plane
- **p** Point on the plane

Definition at line 235 of file Polymesh.java.

7.21.3 Member Data Documentation

7.21.3.1 Vertex obj.Polymesh.Plane.normal = null

Normal vector to the plane.

Definition at line 225 of file Polymesh.java.

Referenced by obj.Polymesh.intersectLinePlane().
7.21.3.2 Vertex obj.Polymesh.Plane.point = null
Point on the plane.
Definition at line 226 of file Polymesh.java.
Referenced by obj.Polymesh.intersectLinePlane().
The documentation for this class was generated from the following file:

- Polymesh.java

7.22 obj.Polymesh.Tri Class Reference

A triangle in 3D space.
Collaboration diagram for obj.Polymesh.Tri:

```
+------------------+
| obj.Polymesh.Tri |
| v                |
| obj.Polymesh.Vertex |
```

Public Methods

- Tri (Vertex a, Vertex b, Vertex c)
  Create a new Triangle from three vertices.

- String toString ()
  Return a string representing the triangle.

Public Attributes

- Vertex[] v = new Vertex[3]
  Vertex (p. 122) list for the triangle.

7.22.1 Detailed Description

A triangle in 3D space.
Definition at line 189 of file Polymesh.java.
7.22.2 Constructor & Destructor Documentation

7.22.2.1 obj.Polymesh.Tri.Tri (Vertex a, Vertex b, Vertex c) [inline]
Create a new Triangle from three vertices.

Parameters:
  a First vertex
  b Second vertex
  c Third vertex

Definition at line 199 of file Polymesh.java.

7.22.3 Member Function Documentation

7.22.3.1 String obj.Polymesh.Tri.toString () [inline]
Return a string representing the triangle.

Returns:
  A string of the form "p1: vertex1.toString() p2: vertex2.toString() p3: vertex3.toString()"

Definition at line 212 of file Polymesh.java.
References obj.Polymesh.Vertex.getIndex().

7.22.4 Member Data Documentation

7.22.4.1 Vertex [] obj.Polymesh.Tri,v = new Vertex[3]
Vertex (p.122) list for the triangle.
Definition at line 190 of file Polymesh.java.
Referenced by obj.Polymesh.edgeify().
The documentation for this class was generated from the following file:

* Polymesh.java

7.23 obj.Polymesh.Vertex Class Reference

A vertex in 3D space using homogeneous coordinates.
Public Methods

- **Vertex** (double x, double y, double z)
  
  Create a new vertex at the specified point.

- **Vertex** (double x, double y, double z, double w)
  
  Create a new vertex at the specified point.

- **String toString ()**
  
  Return a string representing the vertex.

- **void setIndex (int i)**

- **int getIndex ()**

Public Attributes

- **double[] coords = new double[4]**
  
  Coordinates for the vertex.

- **int index**
  
  Index in vertex array.

### 7.23.1 Detailed Description

A vertex in 3D space using homogeneous coordinates.

Definition at line 59 of file Polymesh.java.

### 7.23.2 Constructor & Destructor Documentation

#### 7.23.2.1 obj.Polymesh.Vertex.Vertex (double x, double y, double z) [inline]

Create a new vertex at the specified point.

The vertex's $w$ coordinate is set to 1.

**Parameters:**

- **x** $x$ coordinate
- **y** $y$ coordinate
- **z** $z$ coordinate

Definition at line 71 of file Polymesh.java.

7.23.2.2  obj.PolyMesh.Vertex (double x, double y, double z, double w) [inline]
Create a new vertex at the specified point.

Parameters:
  x $x$ coordinate
  y $y$ coordinate
  z $z$ coordinate
  w $w$ coordinate

Definition at line 86 of file PolyMesh.java.

7.23.3 Member Function Documentation

7.23.3.1 int obj.PolyMesh.Vertex.getIndex () [inline]
Definition at line 107 of file PolyMesh.java.
Referenced by obj.PolyMesh.Tri.toString(), and obj.PolyMesh.Line.toString().

7.23.3.2 void obj.PolyMesh.Vertex.setIndex (int i) [inline]
Definition at line 103 of file PolyMesh.java.
Referenced by obj.PolyMesh.PolyMesh().

7.23.3.3 String obj.PolyMesh.Vertex.toString () [inline]
Return a string representing the vertex.

Returns:
  A string of the form "$x$-coordinate $y$-coordinate $z$-coordinate"

Definition at line 99 of file PolyMesh.java.

7.23.4 Member Data Documentation

Coordinates for the vertex.
Definition at line 60 of file Polymesh.java.
Referenced by obj.Polymesh.HullSort.compare(), obj.Polymesh.dot(),
obj.Polymesh.vertMult(), and obj.Polymesh.vertSub().

7.23.4.2  int obj.Polymesh.Vertex.index
Index in vertex array.
Definition at line 61 of file Polymesh.java.
Referenced by obj.Polymesh.Vertex.getIndex(), and obj.Polymesh.Vertex.set-
Index().
The documentation for this class was generated from the following file:

• Polymesh.java

7.24  disp.Render Class Reference
Holds data needed to render a view.

Public Methods

• int getCurLayer ()
  Get the current layer.

• void setCurLayer (int l)
  Set the current layer.

• void moveUp ()
  Move up one layer if possible.

• void moveDown ()
  Move down one layer if possible.

Private Attributes

• int curLayer = Scene.curScene.getLayers()/2
  < Indicates the current layer in this rendering of the scene

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.24 Detailed Description

Holds data needed to render a view.
Definition at line 37 of file Render.java.

7.24.2 Member Function Documentation

7.24.2.1 int disp.Render.getCurLayer () [inline]
Get the current layer.

Returns:
current layer

Definition at line 49 of file Render.java.
References disp.Render.curlayer.
Referenced by disp.Scene.drawScene().

7.24.2.2 void disp.Render.moveDown () [inline]
Move down one layer if possible.
Definition at line 82 of file Render.java.
References disp.Render.curlayer.

7.24.2.3 void disp.Render.moveUp () [inline]
Move up one layer if possible.
Definition at line 71 of file Render.java.
References disp.Render.curlayer.

7.24.2.4 void disp.Render.setCurLayer (int l) [inline]
Set the current layer.

Parameters:
l current layer

Definition at line 58 of file Render.java.
References disp.Render.curlayer.
7.24.3 Member Data Documentation

7.24.3.1 int disp.Render.curlayer = Scene.curScene.getLayers()/2
[private]
< Indicates the current layer in this rendering of the scene
Definition at line 40 of file Render.java.
Referenced by disp.Render.getCurLayer(), disp.Render.moveDown(),
disp.Render.moveUp(), and disp.Render.setCurLayer().
The documentation for this class was generated from the following file:

- Render.java

7.25 disp.Scene Class Reference

This holds all the information about a scene.
Inheritance diagram for disp.Scene:

```
io.ParserInterface
   `-- disp.Scene
```

Collaboration diagram for disp.Scene:

```
io.ParserInterface
       `-- curScene
```

Public Methods

- Scene ()
  Creates a new scene.

- void initScene ()
  Initialize the scene.
- void **showColorChooserDialog** (Component c)
  
- void **selectObject** (Point pnt, **PickablePolygon**[] objList)
  
  *Finds the object selected by a mouse click.*

- void **drawScene** (Graphics graph, Rectangle r, **Render** rv, Vector objListReference)

  *Draw current scene.*

- void **drawSelectionMarkers** (Graphics2D g, SceneGraphNode sgn)

  *Draws small black selection markers around the currently selected polygon.*

- void **clearScene3D** ()

  *Clear the current branch group, clearing the scene displayed in J3DWindows.*

- Group **drawScene3D** ()

  *This is the 3D analog to drawScene() (p. 132).*

- void **add** (SceneGraphNode object)

  *Add an object to the list.*

- boolean **deleteSelectedObject** ()

  *Delete the currently selected object, if an object is selected.*

- void **setFilename** (File file)

  *Filename used in the scene.*

- File **getFilename** ()

  *Get the filename.*

- void **setChanged** (boolean c)

  *Set the changed state.*

- boolean **getChanged** ()

  *Get changed state.*

- void **setColor** (Color newColor)

  *Set the current drawing color.*

- Color **getColor** ()

  *Get the current drawing color.*

- SceneGraphNode **getCurrentObject** ()
Get the currently picked object.

- **String toString ()**
  
  Convert the scene to a string.

### Static Public Methods

- **void setDirty ()**
  
  Increments dirty count.

- **int getDirty ()**
  
  Gets the dirty count.

- **void setDirtyMarkers ()**
  
  Sets the dirty markers.

- **int getDirtyMarkers ()**
  
  Gets the dirty markers.

- **void setLayers (int l)**
  
  Change the maximum number of layers.

- **int getLayers ()**
  
  Get the maximum number of layers.

- **void setDrawGridlines (boolean b)**
  
  Set the gridlines.

- **boolean getDrawGridlines ()**
  
  Get the gridlines.

- **void setDrawCalibrationMarks (boolean b)**
  
  Set the calibration marks.

- **boolean getDrawCalibrationMarks ()**
  
  Get the calibration marks.
Static Public Attributes

- Scene `curScene = new Scene();
  
  Global to hold one static scene.
- final int `defaultLayers = 30`

Private Methods

- void `drawGridlines (Graphics2D g)`
  
  Draws gridlines on the graphics object.
- void `drawCalibrationMarks (Graphics2D g)`
  
  This function draws the calibration marks.

Private Attributes

- File `filename = null`
  
  Current filename.
- boolean `changed = false`
  
  Has scene been changed.
- SceneGraphNode `currentObject = null`
  
  Currently selected object.
- Color `drawingColor = new Color(255, 50, 50)`
  
  Current drawing color.
- SceneGraph `sceneGraph = null`
  
  Current SceneGraph to display.
- JDialog `colorChooserDialog = null`
- JColorChooser `colorChooser = null`

Static Private Attributes

- int `layers = defaultLayers`
  
  Number of layers.
- int `dCount = 1`
Current dirty counter.

- int dCountMarkers = 0
  Current dirty counter for selection markers.

- boolean drawGrid = true
- boolean drawCalib = true

### 7.25.1 Detailed Description

This holds all the information about a scene.
Definition at line 64 of file Scene.java.

### 7.25.2 Constructor & Destructor Documentation

#### 7.25.2.1 disp.Scene,Scene () [inline]

Creates a new scene.
Definition at line 89 of file Scene.java.
References disp.Scene.initScene().

### 7.25.3 Member Function Documentation

#### 7.25.3.1 void disp.Scene,add (SceneGraphNode object) [inline]

Add an object to the list.

**Parameters:**

- *object* object to add

Implements io.ParserInterface (p.89).
Definition at line 309 of file Scene.java.
References disp.Scene.changed, disp.Scene.sceneGraph, and disp.Scene.setDirty().

#### 7.25.3.2 void disp.Scene,clearScene3D () [inline]

Clear the current branch group, clearing the scene displayed in J3DWindows.
Definition at line 285 of file Scene.java.
Referenced by disp.Scene.initScene().
7.25.3.3  boolean disp.Scene.deleteSelectedObject ()  [inline]
Delete the currently selected object, if an object is selected.
Upon return, no object will be selected.

Returns:
Returns true if an object was selected and successfully deleted. Returns false otherwise.

Definition at line 322 of file Scene.java.
References disp.Scene.changed, disp.Scene.currentObject, disp.Scene.sceneGraph, and disp.Scene.setDirty().

7.25.3.4  void disp.Scene.drawCalibrationMarks (Graphics2D g)  [inline, private]
This function draws the calibration marks.

Parameters:
  g Graphics to draw on.

Definition at line 233 of file Scene.java.
Reference by disp.Scene.drawScene().

7.25.3.5  void disp.Scene.drawGridlines (Graphics2D g)  [inline, private]
Draws gridlines on the graphics object.

Parameters:
  g Graphics object on which to draw gridlines

Definition at line 194 of file Scene.java.
Reference by disp.Scene.drawScene().

7.25.3.6  void disp.Scene.drawScene (Graphics graph, Rectangle r, Render rv, Vector objListReference)  [inline]
Draw current scene.

Parameters:
  graph device to draw on
  r min/max range
rv render options to use

Definition at line 147 of file Scene.java.


7.25.3.7 Group disp.Scene,drawScene3D () [inline]
This is the 3D analog to drawScene() (p. 132).
It returns a BranchGroup that represents the current scene.
Definition at line 300 of file Scene.java.
References disp.Scene.sceneGraph.

7.25.3.8 void disp.Scene,drawSelectionMarkers (Graphics2D g, SceneGraphNode sgn) [inline]
Draws small black selection markers around the currently selected polygon.

Parameters:
  g device to draw on
  sgn Reference for the polygon back into the scene graph

Definition at line 258 of file Scene.java.
Referenced by disp.Scene.drawScene().

7.25.3.9 boolean disp.Scene,getC heaned () [inline]
Get changed state.
This indicates we need to save.

Returns:
  need to save

Definition at line 394 of file Scene.java.
Referenced by disp.Scene.changed.

7.25.3.10 Color disp.Scene.getColor () [inline]
Get the current drawing color.
7.25.3.11 SceneGraphNode disp.Scene.getCurrentObject () [inline]
Get the currently picked object.

Returns:
Current picked object

7.25.3.12 int disp.Scene.getDirty () [inline, static]
 Gets the dirty count.

Returns:
Returns the dirty count

7.25.3.13 int disp.Scene.getDirtyMarkers () [inline, static]
 Gets the dirty markers.

Returns:
Dirty?

7.25.3.14 boolean disp.Scene.getDrawCalibrationMarks () [inline, static]
Get the calibration marks.
Returns:
calibration marks

Definition at line 442 of file Scene.java.
References disp.Scene.drawCalib.

7.25.3.15 boolean disp.Scene.getDrawGridlines () [inline, static]
Get the gridlines.

Returns:
gridlines

Definition at line 427 of file Scene.java.
References disp.Scene.drawGrid.

7.25.3.16 File disp.Scene.getFilename () [inline]
Get the filename.

Returns:
filename

Definition at line 346 of file Scene.java.
References disp.Scene.filename.

7.25.3.17 int disp.Scene.getLayers () [inline, static]
Get the maximum number of layers.

Returns:
max layers

Definition at line 412 of file Scene.java.
References disp.Scene.layers.

7.25.3.18 void disp.Scene.initScene () [inline]
Initialize the scene.

By making this a separate public method, we can discard the current scene and
start a new one without having to make a new instance of the Scene (p.127)
class.

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
Definition at line 98 of file Scene.java.
References disp.Scene.clearScene3D(), disp.Scene.colorChooser, disp.Scene.colorChooserDialog, disp.Scene.sceneGraph, and disp.Scene.setColor().
Referenced by disp.Scene.Scene().

7.25.3.19 void disp.Scene.selectObject (Point *pnt, PickablePolygon *objList[]) [inline]
Finds the object selected by a mouse click.

Parameters:
   *pnt* Coordinate of the mouse click

Definition at line 121 of file Scene.java.
References disp.Scene.currentObject, disp.PickablePolygon.getPolygon(), disp.PickablePolygon.getSceneGraphNode(), and disp.Scene.setDirtyMarkers().

7.25.3.20 void disp.Scene.setChanged (boolean *c) [inline]
Set the changed state.

Parameters:
   *c* have we changed

Definition at line 385 of file Scene.java.
References disp.Scene.changed.

7.25.3.21 void disp.Scene.setColor (Color *newColor) [inline]
Set the current drawing color.

Parameters:
   *newColor* set the new drawing color

Implements io.ParserInterface (p. 89).
Definition at line 451 of file Scene.java.
References disp.Scene.changed, disp.Scene.currentObject, disp.Scene.drawingColor, and disp.Scene.setDirty().
Referenced by disp.Scene.initScene().
7.25.3.22  void disp.Scene.setDirty () [inline, static]
Increments dirty count.
Definition at line 353 of file Scene.java.
References disp.Scene.dCount.
Referenced by disp.Scene.add(), disp.Scene.deleteSelectedObject(), and disp.Scene.setColor().

7.25.3.23  void disp.Scene.setDirtyMarkers () [inline, static]
Sets the dirty markers.
Definition at line 368 of file Scene.java.
References disp.Scene.dCountMarkers.
Referenced by disp.Scene.selectObject().

7.25.3.24  void disp.Scene.setDrawCalibrationMarks (boolean b) [inline, static]
Set the calibration marks.
Definition at line 434 of file Scene.java.
References disp.Scene.drawCalib.

7.25.3.25  void disp.Scene.setDrawGridlines (boolean b) [inline, static]
Set the gridlines.
Definition at line 419 of file Scene.java.
References disp.Scene.drawGrid.

7.25.3.26  void disp.Scene.setFilename (File file) [inline]
Filename used in the scene.

Parameters:
  file  filename to use

Definition at line 337 of file Scene.java.
References disp.Scene.filename.
7.25.3.27 void disp.Scene.setLayers (int l) [inline, static]
Change the maximum number of layers.

Parameters:
   l  new max

Definition at line 403 of file Scene.java.
References disp.Scene.layers.

7.25.3.28 void disp.Scene.showColorChooserDialog (Component c) [inline]
Definition at line 112 of file Scene.java.
References disp.Scene.colorChooserDialog.

7.25.3.29 String disp.Scene.toString () [inline]
Convert the scene to a string.
Definition at line 484 of file Scene.java.
References disp.Scene.drawingColor, and disp.Scene.sceneGraph.

7.25.4 Member Data Documentation

7.25.4.1 boolean disp.Scene.changed = false [private]
Has scene been changed.
Definition at line 72 of file Scene.java.
Referenced by disp.Scene.add(), disp.Scene.deleteSelectedObject(),
disp.Scene.getChanged(), disp.Scene.setSelected(), and disp.Scene.setColor().

7.25.4.2 JColorChooser disp.Scene.colorChooser = null [private]

Definition at line 81 of file Scene.java.
Referenced by disp.Scene.initScene().

7.25.4.3 JDialog disp.Scene.colorChooserDialog = null [private]

Definition at line 80 of file Scene.java.
Referenced by disp.Scene.initScene(), and disp.Scene.showColorChooser-
Dialog().
7.25.4.4 SceneGraphNode disp.Scene.currentObject = null
[private]
Currently selected object.
Definition at line 74 of file Scene.java.
Referenced by disp.Scene.deleteSelectedObject(), disp.Scene.drawScene(),
disp.Scene.getCurrentObject(), disp.Scene.selectObject(), and disp.Scene.set-
Color().

7.25.4.5 Scene disp.Scene.curScene = new Scene() [static]
Global to hold one static scene.
Definition at line 68 of file Scene.java.

7.25.4.6 int disp.Scene.dCount = 1 [static, private]
Current dirty counter.
Definition at line 77 of file Scene.java.
Referenced by disp.Scene.getDirty(), and disp.Scene.setDirty().

7.25.4.7 int disp.Scene.dCountMarkers = 0 [static, private]
Current dirty counter for selection markers.
Definition at line 78 of file Scene.java.
Referenced by disp.Scene.getDirtyMarkers(), and disp.Scene.setDirtyMarkers().

7.25.4.8 final int disp.Scene.defaultLayers = 30 [static]
Definition at line 69 of file Scene.java.

7.25.4.9 boolean disp.Scene.drawCalib = true [static, private]
Definition at line 84 of file Scene.java.
Referenced by disp.Scene.drawScene(), disp.Scene.getDrawCalibrationMarks(),
and disp.Scene.setDrawCalibrationMarks().

7.25.4.10 boolean disp.Scene.drawGrid = true [static, private]
Definition at line 83 of file Scene.java.
Referenced by disp.Scene.drawScene(), disp.Scene.getGridlines(), and
disp.Scene.setDrawGridlines().

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.25.4.11 Color disp.Scene.drawingColor = new Color(255, 50, 50) [private]
Current drawing color.
Definition at line 75 of file Scene.java.
Referenced by disp.Scene.getColor(), disp.Scene.setColor(), and disp.Scene.toString().

7.25.4.12 File disp.Scene.filename = null [private]
Current filename.
Definition at line 71 of file Scene.java.
Referenced by disp.Scene.getFilename(), and disp.Scene.setFilename().

7.25.4.13 int disp.Scene.layers = defaultLayers [static, private]
Number of layers.
Definition at line 73 of file Scene.java.
Referenced by disp.Scene-desscene(), disp.Scene.getLayers(), and disp.Scene.setLayers().

7.25.4.14 SceneGraph disp.Scene,sceneGraph = null [private]
Current SceneGraph to display.
Definition at line 76 of file Scene.java.
Referenced by disp.Scene.add(), disp.Scene.deleteSelectedObject(), disp.Scene-desscene(), disp.Scene-desscene3D(), disp.Scene.initScene(), and disp.Scene.toString().
The documentation for this class was generated from the following file:

- Scene.java

7.26 obj.SceneGraph Class Reference

Controls access to all the SceneGraphNodes.
Collaboration diagram for obj.SceneGraph:
Public Methods

- Group **J3Dscene ()**
  
  Convert the root node into a scene.

- TransformGroup **J3Dscene (SceneGraphNode sg)**
  
  Return a TransformGroup representing the **SceneGraphNode** (p. 145), including its children (done by recursion), and its **Transform** (p. 180).

- **SceneGraph ()**
  
  Create a new, initially empty, **SceneGraph** (p. 140).

- **SceneGraphNode getRoot ()**
  
  Get the root node.

- void **addChild (SceneGraphNode sgn)**
  
  Add a new child to the root node of the **SceneGraph** (p. 140).

- boolean **deleteChild (SceneGraphNode doomedChild)**
  
  Delete the child from the **SceneGraph** (p. 140), if such a child is present.

- PickablePolygon[] **slice** (double z)
  
  Slices all children of the **SceneGraphNode** (p. 145) root.

- String **toString ()**

Static Public Methods

- Shape3D **J3DObj (Polymesh newobj, Color color)**
  
  Return a Shape3D representing the **Polymesh** (p. 107).

- **GeometryArray getGeometryArray (Polymesh newobj)**
This function takes a Polymesh (p. 107) and converts it to a Geometry-Array.

Public Attributes

- **SceneGraphNode root**
  
  Root node of scene graph.

Static Public Attributes

- Hashtable **knownGroups** = null
  
  Hashtable of known tessellations.

7.26.1 Detailed Description

Controls access to all the SceneGraphNode nodes.
Definition at line 56 of file SceneGraph.java.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 `obj.SceneGraph.SceneGraph()` [inline]

Create a new, initially empty, **SceneGraph** (p. 140).
Definition at line 196 of file SceneGraph.java.

7.26.3 Member Function Documentation

7.26.3.1 `void obj.SceneGraph.addChild(SceneGraphNode sgn)` [inline]

Add a new child to the root node of the **SceneGraph** (p. 140).

**Note:**

Maybe we don’t want to provide this method. The same thing could be accomplished by calling SceneGraphObj.getRoot().addChild(foobar). (rdl)

**Parameters:**

- **sgn** child to add.
Definition at line 218 of file SceneGraph.java.
References obj.SceneGraphNode.addChild().

7.26.3.2 boolean obj.SceneGraph.deleteChild (SceneGraphNode doomedChild) [inline]
Delete the child from the SceneGraph (p. 140), if such a child is present.

Note:
Maybe we don’t want to provide this method. See the note on the addChild() (p. 142) method for more info. (rdl)

Parameters:
Child to be removed, if present

Returns :
True if the child was found in the list, and deleted. False otherwise.

Definition at line 233 of file SceneGraph.java.
References obj.SceneGraphNode.deleteChild().

7.26.3.3 GeometryArray obj.SceneGraph.getGeometryArray (Polymesh newobj) [inline, static]
This function takes a Polymesh (p. 107) and converts it to a GeometryArray.

Parameters:
newobj Incoming polymesh.

Definition at line 157 of file SceneGraph.java.
References obj.Polymesh.fList.
Referenced by obj.SceneGraph.J3Dobj().

7.26.3.4 SceneGraphNode obj.SceneGraph.getRoot () [inline]
Get the root node.

Returns :
the root node of the SceneGraph (p. 140)

Definition at line 205 of file SceneGraph.java.
7.26.3.5 Shape3D obj.SceneGraph.J3Dobj (Polymesh newobj, Color color) [inline, static]

Return a Shape3D representing the Polymesh (p. 107).

Note:
Once this routine is finished, we should make it save this Shape3D to a static member variable, since all spheres, cubes, etc. "look" exactly the same. This will save a lot of time when regenerating the Java3D representation of a SceneGraph (p. 140). (rdl)

Returns:
Shape3D representing the Polymesh (p. 107)

Definition at line 112 of file SceneGraph.java.
References obj.SceneGraph.getGeometryArray(), and obj.SceneGraph.knownGroups.
Referenced by obj.SceneGraph.J3Dscene().

7.26.3.6 TransformGroup obj.SceneGraph.J3Dscene (SceneGraphNode sg) [inline]

Return a TransformGroup representing the SceneGraphNode (p. 145), including its children (done by recursion), and its Transform (p. 180).

Returns:
TransformGroup

Definition at line 86 of file SceneGraph.java.
References obj.SceneGraphNode.getChildren(), obj.SceneGraphNode.getColor(), obj.Transform.getMatrix(), obj.SceneGraphNode.getPolymesh(), obj.SceneGraphNode.getTransform(), obj.SceneGraph.J3Dobj(), and obj.SceneGraph.J3Dscene().

7.26.3.7 Group obj.SceneGraph.J3Dscene () [inline]

Convert the root node into a scene.

Returns:
Java3D Group representing the SceneGraph (p. 140).

Definition at line 65 of file SceneGraph.java.
Referenced by obj.SceneGraph.J3Dscene().

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.26.3.8 PickablePolygon [] obj.SceneGraphNode.slice (double z) [inline]
Slices all children of the SceneGraphNode (p.145) root.

Returns:
- Polygons that may be picked

Definition at line 243 of file SceneGraph.java.
References obj.SceneGraphNode.getChildren(), obj.SceneGraphNode.getTransform(), and obj.SceneGraphNode.slice().

7.26.3.9 String obj.SceneGraph.toString () [inline]
Definition at line 256 of file SceneGraph.java.
References obj.SceneGraphNode.getChildren().

7.26.4 Member Data Documentation

7.26.4.1 Hashtable obj.SceneGraph.knownGroups = null [static]
Hashtable of known tessellations.
Definition at line 58 of file SceneGraph.java.
Referenced by obj.SceneGraph.I3Dobj().

7.26.4.2 SceneGraphNode obj.SceneGraph.root
Root node of scene graph.
Definition at line 57 of file SceneGraph.java.
The documentation for this class was generated from the following file:

- SceneGraph.java

7.27 obj.SceneGraphNode Class Reference

Nodes that compose a SceneGraph (p.140).
Collaboration diagram for obj.SceneGraphNode:
Public Methods

- String toString ()
- SceneGraphNode ()
  
  *Construct a new node with no children, no polymesh, and the identity matrix as its transform.*

- SceneGraphNode (Polymesh p, Color c)
  
  *Construct a new node with the the supplied Polymesh (p. 107), no children, and the identity matrix as its transform.*

- void addChild (SceneGraphNode newSG)
  
  *Add a child to the current node.*

- void addTransform (Transform t)
  
  *Append the Transform (p. 180) to the current Transform (p. 180).*

- Transform toOriginTransform ()
- Transform fromOriginTransform ()
- Transform getTransform ()
  
  *Return the current Transform (p. 180).*

- LinkedList getChildren ()
  
  *Return a LinkedList of this node’s children.*

- Polymesh getPolymesh ()
  
  *Return this node’s Polymesh (p. 107).*

- void setPolymesh (Polymesh p)
  
  *Set this node’s Polymesh (p. 107).*
void updateBounds ()
   Update this node’s bounding box, by looking at the bounding box of all its
   children, and its polymesh.

boolean deleteChild (SceneGraphNode doomedChild)
   Delete the child from the SceneGraph (p. 140), if such a child is present.

Color getColor ()
   Return the node’s color.

void setColor (Color c)
   Set the node’s polymesh color (if present).

BoundingBox
Functions to get this node’s bounding box.

Returns :
   x and y coordinates of this node’s bounding box

   • double getMinX ()
   • double getMaxX ()
   • double getMinY ()
   • double getMaxY ()

Public Attributes

• Transform trans
   Transform (p. 180) to apply to poly, and children.

• Polymesh poly
   Polymesh (p. 107) to render (may be null).

• LinkedList children
   List of children.

• Color color
   Color to draw this polymesh (may be null).
Protected Methods

- void slice (Transform t, SceneGraphNode ref, double z, LinkedList polygons)
  
  Add polygons for drawing a slice of this node to the polygons list.

Protected Attributes

- double xMin
  
  Bounding box for this node and its children.

- double yMin
  
  Bounding box for this node and its children.

- double xMax
  
  Bounding box for this node and its children.

- double yMax
  
  Bounding box for this node and its children.

7.27.1 Detailed Description

Nodes that compose a SceneGraph (p. 140).

Each node has a Polymesh (p. 107) and a Transformation, as well as references

to its children.

Definition at line 47 of file SceneGraphNode.java.

7.27.2 Constructor & Destructor Documentation

7.27.2.1 obj,SceneGraphNode,SceneGraphNode () [inline]

Construct a new node with no children, no polymesh, and the identity matrix as its transform.

Definition at line 86 of file SceneGraphNode.java.

References obj.SceneGraphNode.children, and obj.SceneGraphNode.color.

7.27.2.2 obj,SceneGraphNode,SceneGraphNode (Polymesh p, Color c) [inline]
Construct a new node with the supplied \texttt{Polymesh} (p. 107), no children, and the identity matrix as its transform.

\textbf{Parameters:}
- \texttt{p Polymesh} (p. 107) to store in this node
- \texttt{c Color of polymesh p}

Definition at line 100 of file \texttt{SceneGraphNode.java}.
References \texttt{obj.SceneGraphNode.color}.

\textbf{7.27.3 Member Function Documentation}

\textbf{7.27.3.1} \texttt{void obj.SceneGraphNode.addChild \textup{(}SceneGraphNode newSG\textup{)} [inline]}

Add a child to the current node.

\textbf{Parameters:}
- \texttt{newSG Child} to add the current node

Definition at line 111 of file \texttt{SceneGraphNode.java}.
References \texttt{obj.SceneGraphNode.children}.
Referenced by \texttt{obj.SceneGraph.addChild()}.

\textbf{7.27.3.2} \texttt{void obj.SceneGraphNode.addTransform \textup{(}Transform t\textup{)} [inline]}

Append the \texttt{Transform} (p. 180) to the current \texttt{Transform} (p. 180).

\textbf{Parameters:}
- \texttt{t Transform} (p. 180) to append to the current \texttt{Transform} (p. 180)

Definition at line 120 of file \texttt{SceneGraphNode.java}.
References \texttt{obj.Transform.addTransform()}.

\textbf{7.27.3.3} \texttt{boolean obj.SceneGraphNode.deleteChild \textup{(}SceneGraphNode doomedChild\textup{)} [inline]}

Delete the child from the \texttt{SceneGraph} (p. 140), if such a child is present.

\textbf{Parameters:}
- \texttt{Child} to be removed, if present
Returns:
True if the child was found in the list, and deleted. False otherwise.

Definition at line 224 of file SceneGraphNode.java.
References obj.SceneGraphNode.children.
Referenced by obj.SceneGraph.deleteChild().

7.27.3.4 Transform obj.SceneGraphNode,fromOriginTransform () [inline]
Definition at line 134 of file SceneGraphNode.java.
References obj.Transform.apply().

7.27.3.5 LinkedList obj.SceneGraphNode.getChildren () [inline]
Returns:
A LinkedList of this node’s children.
Returns:
This node’s children. The list may be empty.

Definition at line 156 of file SceneGraphNode.java.
References obj.SceneGraphNode.children.
Referenced by obj.SceneGraph.J3Dscene(), obj.SceneGraph.slice(), and
obj.SceneGraph.toString().

7.27.3.6 Color obj.SceneGraphNode.getColor () [inline]
Returns:
The node’s color.

Definition at line 263 of file SceneGraphNode.java.
References obj.SceneGraphNode.color.
Referenced by obj.SceneGraph.J3Dscene().

7.27.3.7 double obj.SceneGraphNode.getMaxX () [inline]
Definition at line 287 of file SceneGraphNode.java.
References obj.SceneGraphNode.xMax.
7.27.3.8 double obj.SceneGraphNode.getMaxY () [inline]
Definition at line 293 of file SceneGraphNode.java.
References obj.SceneGraphNode.yMax.

7.27.3.9 double obj.SceneGraphNode.getMinX () [inline]
Definition at line 284 of file SceneGraphNode.java.
References obj.SceneGraphNode.xMin.

7.27.3.10 double obj.SceneGraphNode.getMinY () [inline]
Definition at line 290 of file SceneGraphNode.java.
References obj.SceneGraphNode.yMin.

7.27.3.11 Polymesh obj.SceneGraphNode.getPolymesh () [inline]
Return this node’s Polymesh (p.107).
(May be null.)

Returns:
This node’s Polymesh (p.107). May by null.

Definition at line 165 of file SceneGraphNode.java.
Referenced by obj.SceneGraph.J3Dscene().

7.27.3.12 Transform obj.SceneGraphNode.getTransform () [inline]
Return the current Transform (p.180).

Returns:
The current Transform (p.180).

Definition at line 147 of file SceneGraphNode.java.
Referenced by obj.SceneGraph.J3Dscene(), and obj.SceneGraph.slice().

7.27.3.13 void obj.SceneGraphNode.setColor (Color c) [inline]
Set the node’s polymesh color (if present).
Parameters:

c The polymesh's new color. If this node does not have a polymesh, but one is added later, this color will be used at that time.

Definition at line 274 of file SceneGraphNode.java.
References obj.SceneGraphNode.color.

7.27.3.14 void obj.SceneGraphNode.setPolymesh (Polymesh p) [inline]

Set this node's Polymesh (p. 107).

Parameters:

p Polymesh (p. 107) to assign to this node. Using null here erases this node's polymesh.

Definition at line 175 of file SceneGraphNode.java.

7.27.3.15 void obj.SceneGraphNode.slice (Transform t, SceneGraphNode ref, double z, LinkedList polygons) [inline, protected]

Add polygons for drawing a slice of this node to the polygons list.

Parameters:

t Transformation matrix from the node's parent
ref Callback if this node is picked in the 2D window
z z-height at which to slice the node
polygons List of polygons that compose the slice

Definition at line 237 of file SceneGraphNode.java.
References obj.Transform.addTransformNew(), obj.SceneGraphNode.children, obj.SceneGraphNode.color, and obj.Polymesh.slice().
Referenced by obj.SceneGraph.slice().

7.27.3.16 Transform obj.SceneGraphNode.toOriginTransform () [inline]

Definition at line 124 of file SceneGraphNode.java.
References obj.Transform.apply().
7.27.3.17  String obj.SceneGraphNode.toString ()  [inline]
Definition at line 58 of file SceneGraphNode.java.
References obj.SceneGraphNode.children, and obj.SceneGraphNode.color.

7.27.3.18  void obj.SceneGraphNode.updateBounds ()  [inline]
Update this node’s bounding box, by looking at the bounding box of all it
children, and its polymesh.
This node must have been sliced for the bounding box to be correct.
Definition at line 185 of file SceneGraphNode.java.
References obj.SceneGraphNode.children, obj.Polymesh.xMax, obj.Scene-
GraphNode.xMax, obj.Polymesh.xMin, obj.SceneGraphNode.xMin,
obj.Polymesh.yMax, obj.SceneGraphNode.yMax, obj.Polymesh.yMin, and
obj.SceneGraphNode.yMin.

7.27.4  Member Data Documentation

7.27.4.1  LinkedList obj.SceneGraphNode.children
List of children.
Definition at line 50 of file SceneGraphNode.java.
Referenced by obj.SceneGraphNode.addChild(), obj.SceneGraphNode.delete-
Child(), obj.SceneGraphNode.getChildren(), obj.SceneGraphNode.Scene-
GraphNode(), obj.SceneGraphNode.slice(), obj.SceneGraphNode.toString(),
and obj.SceneGraphNode.updateBounds().

7.27.4.2  Color obj.SceneGraphNode.color
Color to draw this polymesh (may be null).
Definition at line 51 of file SceneGraphNode.java.
Referenced by obj.SceneGraphNode.getColor(), obj.SceneGraphNode.Scene-
GraphNode(), obj.SceneGraphNode.setColor(), obj.SceneGraphNode.slice(),
and obj.SceneGraphNode.toString().

7.27.4.3  Polymesh obj.SceneGraphNode.poly
Polymesh (p. 107) to render (may be null).
Definition at line 49 of file SceneGraphNode.java.

7.27.4.4  Transform obj.SceneGraphNode.trans
Transform (p.180) to apply to poly, and children.
Definition at line 48 of file SceneGraphNode.java.

7.27.4.5 double obj.SceneGraphNode.xMax [protected]
Bounding box for this node and its children.
Definition at line 55 of file SceneGraphNode.java.
Referenced by obj.SceneGraphNode.getMaxX(), and obj.SceneGraphNode.updateBounds().

7.27.4.6 double obj.SceneGraphNode.xMin [protected]
Bounding box for this node and its children.
Definition at line 53 of file SceneGraphNode.java.
Referenced by obj.SceneGraphNode.getMinX(), and obj.SceneGraphNode.updateBounds().

7.27.4.7 double obj.SceneGraphNode.yMax [protected]
Bounding box for this node and its children.
Definition at line 56 of file SceneGraphNode.java.
Referenced by obj.SceneGraphNode.getMaxY(), and obj.SceneGraphNode.updateBounds().

7.27.4.8 double obj.SceneGraphNode.yMin [protected]
Bounding box for this node and its children.
Definition at line 54 of file SceneGraphNode.java.
Referenced by obj.SceneGraphNode.getMinY(), and obj.SceneGraphNode.updateBounds().

The documentation for this class was generated from the following file:

- SceneGraphNode.java

7.28 io.SimpleCharStream Class Reference

An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (without unicode processing).
Public Methods

- **SimpleCharStream** (java.io.Reader dstream, int startline, int startcolumn, int buffersize)
- **SimpleCharStream** (java.io.Reader dstream, int startline, int startcolumn)
- **SimpleCharStream** (java.io.Reader dstream)
- void **ReInit** (java.io.Reader dstream, int startline, int startcolumn, int buffersize)
- void **ReInit** (java.io.Reader dstream, int startline, int startcolumn)
- void **ReInit** (java.io.Reader dstream)
- **SimpleCharStream** (java.io.InputStream dstream, int startline, int startcolumn, int buffersize)
- **SimpleCharStream** (java.io.InputStream dstream, int startline, int startcolumn)
- **SimpleCharStream** (java.io.InputStream dstream)
- void **ReInit** (java.io.InputStream dstream, int startline, int startcolumn, int buffersize)
- void **ReInit** (java.io.InputStream dstream)
- void **ReInit** (java.io.InputStream dstream, int startline, int startcolumn)

Static Public Methods

- final char **BeginToken** () throws java.io.IOException
- final char **readChar** () throws java.io.IOException
- final int **getColumn** ()
- final int **getLine** ()
- final int **getEndColumn** ()
- final int **getEndLine** ()
- final int **getBeginColumn** ()
- final int **getBeginLine** ()
- final void **backup** (int amount)
- final String **GetImage** ()
- final char[] **GetSuffix** (int len)
- void **Done** ()
- void **adjustBeginLineColumn** (int newLine, int newCol)

  Method to adjust line and column numbers for the start of a token.

  .

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
Static Public Attributes

- final boolean staticFlag = true
- int bufsize
- int available
- int tokenBegin
- int bufpos = -1

Static Private Methods

- final void ExpandBuff (boolean wrapAround)
- final void FillBuff () throws java.io.IOException
- final void UpdateLineColumn (char c)

Static Private Attributes

- int buflines []
- int bufcolumn []
- int column = 0
- int line = 1
- boolean prevCharIsCR = false
- boolean prevCharIsLF = false
- java.io.Reader inputStream
- char[] buffer
- int maxNextCharInd = 0
- int inBuf = 0

7.28.1 Detailed Description

An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (without unicode processing).

Definition at line 9 of file SimpleCharStream.java.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 io.SimpleCharStream.SimpleCharStream (java.io.Reader dstream, int startline, int startcolumn, int buffersize) [inline]

Definition at line 237 of file SimpleCharStream.java.

7.28.2.2 io.SimpleCharStream.SimpleCharStream (java.io.Reader
dstream, int startline, int startcolumn) [inline]
Definition at line 254 of file SimpleCharStream.java.

7.28.2.3 io.SimpleCharStream.SimpleCharStream (java.io.Reader
dstream) [inline]
Definition at line 260 of file SimpleCharStream.java.

7.28.2.4 io.SimpleCharStream.SimpleCharStream (java.io.InputStream
dstream, int startline, int startcolumn, int buffersize)
[inline]
Definition at line 293 of file SimpleCharStream.java.

7.28.2.5 io.SimpleCharStream.SimpleCharStream (java.io.InputStream
dstream, int startline, int startcolumn) [inline]
Definition at line 299 of file SimpleCharStream.java.

7.28.2.6 io.SimpleCharStream.SimpleCharStream (java.io.InputStream
dstream) [inline]
Definition at line 305 of file SimpleCharStream.java.

7.28.3 Member Function Documentation

7.28.3.1 void io.SimpleCharStream.adjustBeginLineColumn (int
newLine, int newCol) [inline, static]
Method to adjust line and column numbers for the start of a token.

Definition at line 360 of file SimpleCharStream.java.
References io.SimpleCharStream.bufcolumn, io.SimpleCharStream.bufline,
io.SimpleCharStream.bupos, io.SimpleCharStream.bufsize, io.SimpleChar-
Stream.column, io.SimpleCharStream.inBuf, io.SimpleCharStream.line, and
io.SimpleCharStream.tokenBegin.

7.28.3.2 final void io.SimpleCharStream.backup (int amount)
[inline, static]
Definition at line 230 of file SimpleCharStream.java.

Referenced by io.SimpleCharStream.FillBuff(), and io.ParserTokenManager.getNextToken().

7.28.3.3 final char io.SimpleCharStream.BeginToken () throws java.io.IOException [inline, static]
Definition at line 126 of file SimpleCharStream.java.
Referenced by io.ParserTokenManager.getNextToken().

7.28.3.4 void io.SimpleCharStream.Done () [inline, static]
Definition at line 350 of file SimpleCharStream.java.

7.28.3.5 final void io.SimpleCharStream.ExpandBuff (boolean wrap-Around) [inline, static, private]
Definition at line 31 of file SimpleCharStream.java.
Referenced by io.SimpleCharStream.FillBuff().

7.28.3.6 final void io.SimpleCharStream.FillBuff () throws java.io.IOException [inline, static, private]
Definition at line 81 of file SimpleCharStream.java.
Referenced by io.SimpleCharStream.readChar().

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.28.3.7 final int io.SimpleCharStream.getBeginColumn () [inline, static]
Definition at line 222 of file SimpleCharStream.java.
Referenced by io.ParserTokenManager.jjFillToken().

7.28.3.8 final int io.SimpleCharStream.getBeginLine () [inline, static]
Definition at line 226 of file SimpleCharStream.java.
Referenced by io.ParserTokenManager.jjFillToken().

7.28.3.9 final int io.SimpleCharStream.getColumn () [inline, static]
Deprecated:

See also:
    getColumn (p.159)
Definition at line 201 of file SimpleCharStream.java.

7.28.3.10 final int io.SimpleCharStream.getEndColumn () [inline, static]
Definition at line 214 of file SimpleCharStream.java.
Referenced by io.ParserTokenManager.getNextToken(), and io.ParserTokenManager.jjFillToken().

7.28.3.11 final int io.SimpleCharStream.getLine () [inline, static]
Definition at line 218 of file SimpleCharStream.java.
7.28.3.12 final String io.SimpleCharStream.getImage () [inline, static]
Definition at line 325 of file SimpleCharStream.java.
Referenced by io.ParserTokenManager.getNextToken(), and io.ParserTokenManager.jjFillToken().

7.28.3.13 final int io.SimpleCharStream.getLine () [inline, static]

Deprecated:

See also:
getEndLine (p.159)

Definition at line 210 of file SimpleCharStream.java.

7.28.3.14 final char [] io.SimpleCharStream.getSuffix (int len) [inline, static]
Definition at line 334 of file SimpleCharStream.java.

7.28.3.15 final char io.SimpleCharStream.readChar () throws java.io.IOException [inline, static]
Definition at line 175 of file SimpleCharStream.java.
7.28.16 void io,SimpleCharStream,ReInit (java,io,InputStream

dstream, int startline, int startcolumn) [inline]

Definition at line 320 of file SimpleCharStream.java.
References io,SimpleCharStream,ReInit().

7.28.17 void io,SimpleCharStream,ReInit (java,io,InputStream
dstream) [inline]

Definition at line 316 of file SimpleCharStream.java.
References io,SimpleCharStream,ReInit().

7.28.18 void io,SimpleCharStream,ReInit (java,io,InputStream
dstream, int startline, int startcolumn, int buffsize) [inline]

Definition at line 310 of file SimpleCharStream.java.
References io,SimpleCharStream,ReInit().

7.28.19 void io,SimpleCharStream,ReInit (java,io,Reader
dstream) [inline]

Definition at line 289 of file SimpleCharStream.java.
References io,SimpleCharStream,ReInit().

7.28.20 void io,SimpleCharStream,ReInit (java,io,Reader
dstream, int startline, int startcolumn) [inline]

Definition at line 283 of file SimpleCharStream.java.
References io,SimpleCharStream,ReInit().

7.28.21 void io,SimpleCharStream,ReInit (java,io,Reader
dstream, int startline, int startcolumn, int buffsize) [inline]

Definition at line 264 of file SimpleCharStream.java.

Referenced by `io.SimpleCharStream.ReInit()`, and `io(Parser).ReInit()`.

### 7.28.3.22 final void `io.SimpleCharStream.UpdateLineColumn` (char c) [inline, static, private]

Definition at line 135 of file `SimpleCharStream.java`.


Referenced by `io.SimpleCharStream.readChar()`.

### 7.28.4 Member Data Documentation

#### 7.28.4.1 int `io.SimpleCharStream.available` [static]

Definition at line 13 of file `SimpleCharStream.java`.


#### 7.28.4.2 int `io.SimpleCharStream.bufcolumn[]` [static, private]

Definition at line 17 of file `SimpleCharStream.java`.


#### 7.28.4.3 char [] `io.SimpleCharStream.buffer` [static, private]

Definition at line 27 of file `SimpleCharStream.java`.

Referenced by `io.SimpleCharStream.Done()`, `io.SimpleCharStream.Expand-
7.28.4.4 int io,SimpleCharStream.bufline[] [static, private]
Definition at line 16 of file SimpleCharStream.java.

7.28.4.5 int io,SimpleCharStream.bufpos = -1 [static]
Definition at line 15 of file SimpleCharStream.java.

7.28.4.6 int io,SimpleCharStream.bufsize [static]
Definition at line 12 of file SimpleCharStream.java.

7.28.4.7 int io,SimpleCharStream.column = 0 [static, private]
Definition at line 19 of file SimpleCharStream.java.
Referenced by io,SimpleCharStream.adjustBeginLineColumn(), io,SimpleCharStream.ReInit(), io,SimpleCharStream.SimpleCharStream(), and io,SimpleCharStream.UpdateLineColumn().

7.28.4.8 int io,SimpleCharStream.inBuf = 0 [static, private]
Definition at line 29 of file SimpleCharStream.java.

7.28.4.9 java.io.Reader io.SimpleCharStream.inputStream
[static, private]
Definition at line 25 of file SimpleCharStream.java.

7.28.4.10 int io.SimpleCharStream.line = 1 [static, private]
Definition at line 20 of file SimpleCharStream.java.

7.28.4.11 int io.SimpleCharStream.maxNextCharInd = 0 [static, private]
Definition at line 28 of file SimpleCharStream.java.

7.28.4.12 boolean io.SimpleCharStream.prevCharIsCR = false [static, private]
Definition at line 22 of file SimpleCharStream.java.
Referenced by io.SimpleCharStream.ReInit(), and io.SimpleCharStream.UpdateLineColumn().

7.28.4.13 boolean io.SimpleCharStream.prevCharIsLF = false [static, private]
Definition at line 23 of file SimpleCharStream.java.
Referenced by io.SimpleCharStream.ReInit(), and io.SimpleCharStream.UpdateLineColumn().

7.28.4.14 final boolean io.SimpleCharStream.staticFlag = true [static]
Definition at line 11 of file SimpleCharStream.java.

7.28.4.15  int io.SimpleCharStream.tokenBegin  [static]
Definition at line 14 of file SimpleCharStream.java.
The documentation for this class was generated from the following file:

- SimpleCharStream.java

7.29  io.SpectreFileError Class Reference

Exception used by the parser to indicate file errors encountered while parsing a file.

Public Methods

- SpectreFileError (String message)
  The only constructor for the exception.

7.29.1  Detailed Description

Exception used by the parser to indicate file errors encountered while parsing a file.
The parser can also return ParseException (p.72) errors, which come from code generated by JavaCC.

Note:

It might be a good idea to catch all ParseExceptions, and stick their messages into a SpectreFileError (p.165). That way, any code external to the parser only sees one type of exception. (rdl)

Definition at line 45 of file SpectreFileError.java.
7.29.2 Constructor & Destructor Documentation

7.29.2.1 \texttt{io.SpectreFileError.SpectreFileError} (String \textit{message})
\texttt{[inline]}

The only constructor for the exception.

\textbf{Parameters:}

\textit{message} String to use as the exception’s error message.

Definition at line 51 of file \texttt{SpectreFileError.java}.

The documentation for this class was generated from the following file:

- \texttt{SpectreFileError.java}

7.30 \texttt{win.SplashWindow} Class Reference

Public Methods

- \texttt{SplashWindow} ()

\textit{Setup the window, then display it.}

7.30.1 Constructor & Destructor Documentation

7.30.1.1 \texttt{win.SplashWindow.SplashWindow} () \texttt{[inline]}

Setup the window, then display it.

Definition at line 56 of file \texttt{SplashWindow.java}.

The documentation for this class was generated from the following file:

- \texttt{SplashWindow.java}

7.31 \texttt{win.StatusBar} Class Reference

Class for status bar; denotes slice viewed.

Inheritance diagram for \texttt{win.StatusBar}: 

---

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
Collaboration diagram for `win.StatusBar`:

Public Methods

- `StatusBar()`  
  Constructor for status bar.

- `void setLayerPane(String input)`  
  Sets text for status bar.

- `void setColorPane(Color color)`  
  Sets color for status bar.

Private Attributes

- `StatusPane layerPane = new StatusPane("")`  
  Pane for status bar.

- `ColorPane colorPane = new ColorPane(mill)`  
  Color pane for status bar.

7.31.1 Detailed Description

Class for status bar; denotes slice viewed.
Definition at line 42 of file StatusBar.java.
7.31.2 Constructor & Destructor Documentation

7.31.2.1 win.StatusBar.StatusBar () [inline]
Constructor for status bar.
Definition at line 49 of file StatusBar.java.
References win.StatusBar.setLayerPane().

7.31.3 Member Function Documentation

7.31.3.1 void win.StatusBar.setColorPane (Color color) [inline]
Sets color for status bar.
Parameters:
    color Color to display in status bar.
Definition at line 71 of file StatusBar.java.
References win.StatusBar.ColorPane.setColor().
Referenced by win.ControlWindow.updateWindow().

7.31.3.2 void win.StatusBar.setLayerPane (String input) [inline]
Sets text for status bar.
Parameters:
    input Text to display in status bar.
Definition at line 62 of file StatusBar.java.
Referenced by win.StatusBar.StatusBar(), and win.ControlWindow.updateWindow().

7.31.4 Member Data Documentation

7.31.4.1 ColorPane win.StatusBar.colorPane = new ColorPane(null) [private]
Color pane for status bar.
Definition at line 44 of file StatusBar.java.

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.31.4.2 StatusPane  win.StatusBar.layerPane = new StatusPane(""") [private]
Pane for status bar.
Definition at line 43 of file StatusBar.java.
The documentation for this class was generated from the following file:

- StatusBar.java

7.32 win.StatusBar.ColorPane Class Reference

Inheritance diagram for win.StatusBar.ColorPane:

```
<table>
<thead>
<tr>
<th>JPanel</th>
</tr>
</thead>
<tbody>
<tr>
<td>win.StatusBar.ColorPane</td>
</tr>
</tbody>
</table>
```

Collaboration diagram for win.StatusBar.ColorPane:

```
<table>
<thead>
<tr>
<th>JPanel</th>
</tr>
</thead>
<tbody>
<tr>
<td>win.StatusBar.ColorPane</td>
</tr>
</tbody>
</table>
```

Public Methods

- void paintComponent (Graphics graph)
  Display (p. 31) the active color in the status bar.

- ColorPane (Color c)
  Constructor to set color.

- void setColor (Color c)
  Set new color.

Public Attributes

- Color color
7.32.1 Constructor & Destructor Documentation

7.32.1.1 win.StatusBar.ColorPane.ColorPane (Color c) [inline]
Constructor to set color.

Parameters:
  c Default color.

Definition at line 120 of file StatusBar.java.

7.32.2 Member Function Documentation

7.32.2.1 void win.StatusBar.ColorPane.paintComponent (Graphics graph) [inline]
Display (p. 31) the active color in the status bar.

Parameters:
  graph Graphic to draw on.

Definition at line 106 of file StatusBar.java.

7.32.2.2 void win.StatusBar.ColorPane.setColor (Color c) [inline]
Set new color.

Parameters:
  c New color.

Definition at line 130 of file StatusBar.java.
Referenced by win.StatusBar.setColorPane().

7.32.3 Member Data Documentation

7.32.3.1 Color win.StatusBar.ColorPane.color
Definition at line 99 of file StatusBar.java.

The documentation for this class was generated from the following file:

- `StatusBar.java`

### 7.33 win.StatusBar.StatusPane Class Reference

Panel for display in status bar.

Inheritance diagram for win.StatusBar.StatusPane:

![Inheritance Diagram](image1)

Collaboration diagram for win.StatusBar.StatusPane:

![Collaboration Diagram](image2)

**Public Methods**

- **StatusBar**(String text)  
  *Constructor for pane.*

**Private Attributes**

- Font **paneFont** = new Font("Serif", Font.PLAIN, 10)  
  *Font used.*

### 7.33.1 Detailed Description

Panel for display in status bar.

Definition at line 80 of file StatusBar.java.
7.34.1 Constructor & Destructor Documentation

7.34.2.1 win.StatusBar.StatusPane.StatusPane (String text) [inline]

Constructor for pane.

Parameters:

- **text**: Text to display in pane.

Definition at line 88 of file StatusBar.java.

References win.StatusBar.StatusPane-paneFont.

7.34.3 Member Data Documentation

7.34.3.1 Font win.StatusBar.StatusPane-paneFont = new Font("Serif", Font.PLAIN, 10) [private]

Font used.

Definition at line 81 of file StatusBar.java.

Referenced by win.StatusBar.StatusPane.StatusPane().

The documentation for this class was generated from the following file:

- StatusBar.java

### 7.34 io.Token Class Reference

Describes the input token stream.

Collaboration diagram for io.Token:

```
  ioToken ~ next
specialToken
```

Public Methods

- final String **toString** ()

  *Returns the image.*
Static Public Methods

- final Token **newToken**(int ofKind)
  
  Returns a new **Token** (p. 172) object, by default.

Public Attributes

- int **kind**
  
  An integer that describes the kind of this token.

- int **beginLine**
  
  beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.

- int **beginColumn**
  
  beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.

- int **endLine**
  
  beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.

- int **endColumn**
  
  beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.

- String **image**
  
  The string image of the token.

- Token **next**
  
  A reference to the next regular (non-special) token from the input stream.

- Token **specialToken**
  
  This field is used to access special tokens that occur prior to this token, but after the immediately preceding regular (non-special) token.
7.34.1 Detailed Description

Describes the input token stream.
Definition at line 8 of file Token.java.

7.34.2 Member Function Documentation

7.34.2.1 final Token io.Token.newToken (int ofKind) [inline, static]

Returns a new Token (p. 172) object, by default. However, if you want, you can create and return subclass objects based on the value of ofKind. Simply add the cases to the switch for all those special cases. For example, if you have a subclass of Token (p. 172) called IDToken that you want to create if ofKind is ID, simply add something like:

case MyParserConstants.ID : return new IDToken();

to the following switch statement. Then you can cast matchedToken variable to the appropriate type and use it in your lexical actions.

Definition at line 73 of file Token.java.
Referenced by io.ParserTokenManager.jjFillToken().

7.34.2.2 final String io.Token.toString () [inline]

Returns the image.

Definition at line 56 of file Token.java.
References io.Token.image.

7.34.3 Member Data Documentation

7.34.3.1 int io.Token.beginColumn

beginLine and beginColumn describe the position of the first character of this token; endLine and endColumn describe the position of the last character of this token.

Definition at line 22 of file Token.java.


7.34.3.2 int io.Token.beginLine
beginLine and beginColumn describe the position of the first character of this
token; endLine and endColumn describe the position of the last character of
this token.
Definition at line 22 of file Token.java.
Referenced by io.ParsingException.getMessage(), io.ParserTokenManager.jjFill-
Token(), and io.Parser.parseCurrentFile().

7.34.3.3 int io.Token.endColumn

beginLine and beginColumn describe the position of the first character of this
token; endLine and endColumn describe the position of the last character of
this token.
Definition at line 22 of file Token.java.
Referenced by io.ParserTokenManager.jjFillToken().

7.34.3.4 int io.Token.endLine

beginLine and beginColumn describe the position of the first character of this
token; endLine and endColumn describe the position of the last character of
this token.
Definition at line 22 of file Token.java.
Referenced by io.ParserTokenManager.jjFillToken().

7.34.3.5 String io.Token.image

The string image of the token.
Definition at line 27 of file Token.java.
Referenced by io.Parser.colorArgList(), io.Parser.getDrawingColor(),
io.ParsingException.getMessage(), io.ParserTokenManager.jjFillToken(),
and io.Parser.transformArgList().

7.34.3.6 int io.Token.kind

An integer that describes the kind of this token.
This numbering system is determined by JavaCCParser, and a table of these
numbers is stored in the file ...Constants.java.
Definition at line 15 of file Token.java.
Referenced by io_PARSER.Exception.getMessage(), io.Parser.jjConsume_Token(),

7.34.3.7 Token io.Token.next
A reference to the next regular (non-special) token from the input stream.
If this is the last token from the input stream, or if the token manager has not read tokens beyond this one, this field is set to null. This is true only if this token is also a regular token. Otherwise, see below for a description of the contents of this field.
Definition at line 37 of file Token.java.
Referenced by io.ParseException.getMessage(), io.Parser.getNextToken(),

7.34.3.8 Token io.Token.specialToken
This field is used to access special tokens that occur prior to this token, but after the immediately preceding regular (non-special) token.
If there are no such special tokens, this field is set to null. When there are more
than one such special token, this field refers to the last of these special tokens,
which in turn refers to the next previous special token through its specialToken
field, and so on until the first special token (whose specialToken field is null).
The next fields of special tokens refer to other special tokens that immediately
follow it (without an intervening regular token). If there is no such token, this
field is null.
Definition at line 51 of file Token.java.
The documentation for this class was generated from the following file:

- Token.java

7.35 io.TokenMgrError Class Reference

Public Methods

- String getMessage ()
  
  You can also modify the body of this method to customize your error messages.

- TokenMgrError ()
- TokenMgrError (String message, int reason)
- TokenMgrError (boolean EOFSeen, int lexState, int errorLine, int
errorColumn, String errorAfter, char curChar, int reason)
Public Attributes

- int errorCode
  Indicates the reason why the exception is thrown.

Static Public Attributes

- final int LEXICAL_ERROR = 0
  Lexical error occurred.

- final int STATIC_LEXER_ERROR = 1
  An attempt was made to create a second instance of a static token manager.

- final int INVALID_LEXICAL_STATE = 2
  Tried to change to an invalid lexical state.

- final int LOOP_DETECTED = 3
  Detected (and bailed out of) an infinite loop in the token manager.

Static Protected Methods

- final String addEscapes (String str)
  Replaces unprintable characters by their escaped (or unicode escaped) equivalents in the given string.

Static Private Methods

- final String LexicalError (boolean EOFSeen, int lexState, int errorLine, int errorColumn, String errorAfter, char curChar)
  Returns a detailed message for the Error when it is thrown by the token manager to indicate a lexical error.

7.35.1 Constructor & Destructor Documentation

7.35.1.1 io.TokenMgrError,TokenMgrError () [inline]
Definition at line 122 of file TokenMgrError.java.
7.35.1.2 \texttt{io.TokenMgrError.TokenMgrError} (String \textit{message}, int \textit{reason}) [inline]

Definition at line 125 of file TokenMgrError.java.
References \texttt{io.TokenMgrError.errorCode}.

7.35.1.3 \texttt{io.TokenMgrError.TokenMgrError} (boolean \textit{EOFSeen}, int \textit{lexState}, int \textit{errorLine}, int \textit{errorColumn}, String \textit{errorAfter}, char \textit{curChar}, int \textit{reason}) [inline]

Definition at line 130 of file TokenMgrError.java.
References \texttt{io.TokenMgrError.LexicalError()}.

7.35.2 Member Function Documentation

7.35.2.1 final String \texttt{io.TokenMgrError.addEscapes} (String \textit{str}) [inline, static, protected]

Replaces unprintable characters by their spaced (or unicode escaped) equivalents in the given string.

Definition at line 40 of file TokenMgrError.java.
Referenced by \texttt{io.TokenMgrError.LexicalError()}. 

7.35.2.2 String \texttt{io.TokenMgrError.getMessage} () [inline]

You can also modify the body of this method to customize your error messages. For example, cases like \texttt{LOOP\_DETECTED} and \texttt{INVALID\_LEXICAL\_STATE} are not of end-users concern, so you can return something like :

"Internal Error : Please file a bug report .... "

from this method for such cases in the release version of your parser.

Definition at line 114 of file TokenMgrError.java.

7.35.2.3 final String \texttt{io.TokenMgrError.LexicalError} (boolean \textit{EOFSeen}, int \textit{lexState}, int \textit{errorLine}, int \textit{errorColumn}, String \textit{errorAfter}, char \textit{curChar}) [inline, static, private]

Returns a detailed message for the Error when it is thrown by the token manager to indicate a lexical error.

Parameters: EOFSeen : indicates if EOF caused the lexical error curLexState : lexical state in which this error occurred errorLine : line number when the error occurred errorColumn : column number when the error occurred errorAfter

---

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
7.35  io.TokenMgrError Class Reference

: prefix that was seen before this error occurred
: char: the offending character
: Note: You can customize the lexical error message by modifying this method.

Definition at line 97 of file TokenMgrError.java.

References io.TokenMgrError.addEscapes().

Referenced by io.TokenMgrError.TokenMgrError().

7.35.3 Member Data Documentation

7.35.3.1 int io.TokenMgrError.errorCode

Indicates the reason why the exception is thrown.

It will have one of the above 4 values.

Definition at line 34 of file TokenMgrError.java.

Referenced by io.TokenMgrError.TokenMgrError().

7.35.3.2 final int io.TokenMgrError.INVALID_LEXICAL_STATE = 2 [static]

Tried to change to an invalid lexical state.

Definition at line 23 of file TokenMgrError.java.

7.35.3.3 final int io.TokenMgrError.LEXICAL_ERROR = 0 [static]

Lexical error occurred.

Definition at line 13 of file TokenMgrError.java.

7.35.3.4 final int io.TokenMgrError.Loop_DETECTED = 3 [static]

Detected (and bailed out of) an infinite loop in the token manager.

Definition at line 28 of file TokenMgrError.java.

7.35.3.5 final int io.TokenMgrError.STATIC_LEXER_ERROR = 1 [static]

An attempt was made to create a second instance of a static token manager.

Definition at line 18 of file TokenMgrError.java.

The documentation for this class was generated from the following file:

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
• TokenMgrError.java

7.36  obj.Transform Class Reference

Wrapper around a 4x4 transformation matrix for 3D homogeneous coordinates.

Public Methods

• String toString ()
• Transform ()
  
  Create a new transformation.

• Transform (Matrix4d matrix)
  
  Create a new Transform (p.180) from the matrix.

• Transform (double m00, double m01, double m02, double m03, double m10, double m11, double m12, double m13, double m20, double m21, double m22, double m23, double m30, double m31, double m32, double m33)
• Matrix4d getMatrix ()
  
  Return the current transformation matrix.

• void addToTransform (Transform t)
  
  Append Transform (p.180) t to the current transform sequence.

• Transform addToTransformNew (Transform t)
  
  Append transform t to the current transform, and return a new Transform (p.180) object representing this sequence.

• Polymesh.Vertex apply (Polymesh.Vertex v)
  
  Apply the transform to Vertex v.

Public Attributes

• Matrix4d tMatrix
  
  4x4 transformation matrix
7.36.1 Detailed Description

Wrapper around a 4x4 transformation matrix for 3D homogeneous coordinates.
Definition at line 42 of file Transform.java.

7.36.2 Constructor & Destructor Documentation

7.36.2.1 obj.Transform.Transform () [inline]
Create a new transformation.
The new transformation is really the identity matrix, so it has no effect on coordinates.
Definition at line 70 of file Transform.java.
References obj.Transform.tMatrix.
Referenced by obj.Transform.addTransformNew().

7.36.2.2 obj.Transform.Transform (Matrix4d matrix) [inline]
Create a new Transform (p.180) from the matrix.
The matrix must be affine, but this method does not guarantee this. Attempting to use a non-affine transformation will cause Java3D to throw a fatal exception.

Parameters:
matrix 4x4 matrix from which to create a new Transform (p.180)

Definition at line 85 of file Transform.java.
References obj.Transform.tMatrix.

7.36.2.3 obj.Transform.Transform (double m00, double m01, double m02, double m03, double m10, double m11, double m12, double m13, double m20, double m21, double m22, double m23, double m30, double m31, double m32, double m33) [inline]
Definition at line 89 of file Transform.java.
References obj.Transform.tMatrix.

7.36.3 Member Function Documentation

7.36.3.1 void obj.Transform.addTransform (Transform t) [inline]
Append **Transform** (p. 180) t to the current transform sequence.

**Parameters:**

- **t Transform** (p. 180) to append to the end of the current transform sequence. t will be the last transform performed.

Definition at line 114 of file Transform.java.

References obj.Transform.getMatrix(), and obj.Transform.tMatrix.

Referenced by obj.SceneGraphNode.addTransform().

### 7.36.3.2 Transform obj.Transform,addTransformNew (Transform t) [inline]

Append transform t to the current transform, and return a new **Transform** (p. 180) object representing this sequence.

The original objects are not changed.

**Parameters:**

- **t Transform** (p. 180) to append to the end of the current transform sequence. t will be the last transformed performed.

**Returns:**

A copy of the current transform, with t appended. The original object is not changed.

Definition at line 131 of file Transform.java.

References obj.Transform.getMatrix(), obj.Transform.tMatrix, and obj.Transform.Transform().

Referenced by obj.SceneGraphNode.slice().

### 7.36.3.3 Polymesh,Vertex obj.Transform,apply (Polymesh,Vertex v) [inline]

Apply the transform to Vertex v.

**Parameters:**

- **v** Vertex to transform

**Returns:**

Copy of v with the transform applied. The original vertex is not changed.

Definition at line 145 of file Transform.java.
References obj.Transform.tMatrix.

Referenced by obj.SceneGraphNode.fromOriginTransform(), obj.SceneGraphNode.toOriginTransform(), and obj.PolyMesh.transformLine().

7.36.3.4 Matrix4d obj.Transform.getMatrix () [inline]
Return the current transformation matrix.

Returns:
4x4 matrix

Definition at line 103 of file Transform.java.
References obj.Transform.tMatrix.

Referenced by obj.Transform.addTransform(), obj.Transform.addTransformNew(), and obj.SceneGraph.J3DScene().

7.36.3.5 String obj.Transform.toString () [inline]
Definition at line 45 of file Transform.java.
References obj.Transform.tMatrix.

7.36.4 Member Data Documentation

7.36.4.1 Matrix4d obj.Transform.tMatrix
4x4 transformation matrix

Definition at line 43 of file Transform.java.

The documentation for this class was generated from the following file:

- Transform.java

7.37 obj.TransformFactory Class Reference

Contains several static methods that return common Transformations.

Static Public Methods

- Transform newScale (double x, double y, double z)
Build a Transform (p. 180) to scale an object by the indicated scale factors along the x, y, and z axes, respectively.

- **Transform newTranslate** (double x, double y, double z)
  Build a Transform (p. 180) to translate an object by the indicated units along each axis.

- **Transform newRotateX** (double theta)
  Build a Transform (p. 180) to rotate an object about the X-axis by the angle theta (in radians).

- **Transform newRotateY** (double theta)
  Build a Transform (p. 180) to rotate an object about the Y-axis by the angle theta (in radians).

- **Transform newRotateZ** (double theta)
  Build a Transform (p. 180) to rotate an object about the Z-axis by the angle theta (in radians).

### 7.37.1 Detailed Description

Contains several static methods that return common Transformations.
Definition at line 42 of file TransformFactory.java.

### 7.37.2 Member Function Documentation

#### 7.37.2.1 Transform obj.TransformFactory,newRotateX (double theta) [inline, static]

Build a Transform (p. 180) to rotate an object about the X-axis by the angle theta (in radians).

Rotations are done about the origin, and the direction is defined by the right hand rule.

**Parameters:**

- **theta** Radians to rotate

**Returns :**

- Transform (p. 180) that performs the rotate operation

Definition at line 89 of file TransformFactory.java.
7.37.2.2 Transform obj:TransformFactory.newRotateY (double theta) [inline, static]
Build a Transform (p.180) to rotate an object about the Y-axis by the angle theta (in radians).
Rotations are done about the origin, and the direction is defined by the right hand rule.

Parameters:
theta Radians to rotate

Returns:
Transform (p.180) that performs the rotate operation

Definition at line 107 of file TransformFactory.java.

7.37.2.3 Transform obj:TransformFactory.newRotateZ (double theta) [inline, static]
Build a Transform (p.180) to rotate an object about the Z-axis by the angle theta (in radians).
Rotations are done about the origin, and the direction is defined by the right hand rule.

Parameters:
theta Radians to rotate

Returns:
Transform (p.180) that performs the rotate operation

Definition at line 124 of file TransformFactory.java.

7.37.2.4 Transform obj:TransformFactory.newScale (double x, double y, double z) [inline, static]
Build a Transform (p.180) to scale an object by the indicated scale factors along the x, y, and z axes, respectively.
Scaling is done about the origin.

Parameters:
x x scale factor
y y scale factor
z z scale factor
Returns:

- Transform (p.180) that performs the scale operation

Definition at line 53 of file TransformFactory.java.

7.37.2.5 Transform obj,TransformFactory,newTranslate (double x, double y, double z) [inline, static]
Build a Transform (p.180) to translate an object by the indicated units along each axis.

Parameters:

- x x translation distance
- y y translation distance
- z z translation distance

Returns:

- Transform (p.180) that performs the translate operation

Definition at line 72 of file TransformFactory.java.

The documentation for this class was generated from the following file:

- TransformFactory.java

8 SPECTRE File Documentation

8.1 ControlWindow.java File Reference

The main SPECTRE window, the slicer, handles user input and displays slices.

Namespaces

- namespace java.awt
- namespace java.awt.print
- namespace javax.swing
- namespace win

8.1.1 Detailed Description

The main SPECTRE window, the slicer, handles user input and displays slices.

Most of the flow control is done here.

11 Sept. 2002
8.2 disp.doxygen File Reference

Author:
Tyler Nielsen <nielsen@colorado.edu>

Date:
2003/04/28 05:30:40

Revision:
1.57

Definition in file ControlWindow.java.

8.2 disp.doxygen File Reference

Namespaces
- namespace disp

8.3 Display.java File Reference

Handles communication between 2D and 3D windows, and provides a wrapper around the 3D client, so Java3D is not a system requirement.

Namespaces
- namespace win

8.3.1 Detailed Description

Handles communication between 2D and 3D windows, and provides a wrapper around the 3D client, so Java3D is not a system requirement.

13 Feb. 2003

Author:
Tyler Nielsen <nielsen@colorado.edu>

Date:
2003/04/25 00:58:12

Revision:
1.15

Definition in file Display.java.
8.4 ExtensionFilter.java File Reference

Handles extension options in save/open dialog box.

Namespaces

- namespace win

8.4.1 Detailed Description

Handles extension options in save/open dialog box.

30 Oct. 2002

Author:

Tyler Nielsen <nielsen@colorado.edu>

Date:

2003/04/14 05:53:01

Revision:

1.6

Definition in file ExtensionFilter.java.

8.5 Geometry.java File Reference

Implements a class consisting of static functions useful for drawing.

Namespaces

- namespace disp

8.5.1 Detailed Description

Implements a class consisting of static functions useful for drawing.

15 Sept 2002

Author:

Ryan D. Lewis <lewisd@ucsub.colorado.edu>

Date:

2003/04/30 20:25:29
Revision:
1.20

Definition in file Geometry.java.

8.6 io.dox File Reference

Namespaces

- namespace io

8.7 J3DWindow.java File Reference

Creates a window for viewing scenes using Java3D.

Namespaces

- namespace com.sun.j3d.utils.behaviors.mouse
- namespace win

8.7.1 Detailed Description

Creates a window for viewing scenes using Java3D.
This is the three-dimensional analog to ControlWindow.
30 Oct. 2002

Author:
Ryan D. Lewis <lewisrd@ucsub.colorado.edu>

Date:
2003/04/30 19:55:50

Revision:
1.20

Definition in file J3DWindow.java.

8.8 mainpage.dox File Reference

8.9 MainWindow.java File Reference

Opens the ControlWindow, whether opened as an applet or by a call to main.
8.10 MenuHandler.java File Reference

Namespaces

- namespace win

8.9.1 Detailed Description

Opens the ControlWindow, whether opened as an applet or by a call to main.
6 Oct. 2002

**Author:**

Tyler Nielsen <nielsent@colorado.edu>

**Date:**

2003/04/30 20:39:03

**Revision:**

1.16

Definition in file MainWindow.java.

8.10 MenuHandler.java File Reference

Handles majority of menu events.

Namespaces

- namespace java,util,zip
- namespace javax.print.attribute
- namespace javax.print.attribute.standard
- namespace win

8.10.1 Detailed Description

Handles majority of menu events.
14 Sept. 2002

**Author:**

Tyler Nielsen <nielset@colorado.edu>

**Date:**

2003/05/04 20:46:10
8.11 obj.doc File Reference

Namespaces

- namespace obj

8.12 ObjDialog.java File Reference

Creates majority of dialog boxes, for on-the-fly object creation as well as for object manipulations.

Namespaces

- namespace win

8.12.1 Detailed Description

Creates majority of dialog boxes, for on-the-fly object creation as well as for object manipulations.
18 Feb. 2003

Author:
Jeremy Garcia <Jeremy.P.Garcia@colorado.edu>

Date:
2003/05/04 21:16:47

Revision:
1.37

Definition in file ObjDialog.java.

8.13 ObjFactory.java File Reference

Routines to generate Polymeshes for common objects.
8.14 OptionsDialog.java File Reference

Namespaces

- namespace obj

8.13.1 Detailed Description

Routines to generate Polymeshes for common objects.
7 Feb. 2003

Author:
Ryan D. Lewis <lewisrd@ncsub.colorado.edu>

Date:
2003/04/30 19:54:38

Revision:
1.20

Definition in file ObjFactory.java.

8.14 OptionsDialog.java File Reference

Dialog box for setting Scene-wide options.

Namespaces

- namespace win

8.14.1 Detailed Description

Dialog box for setting Scene-wide options.
9 Apr. 2003

Author:
Ryan D. Lewis <lewisrd@ncsub.colorado.edu>

Date:
2003/05/04 21:16:47

Revision:
1.6

Definition in file OptionsDialog.java.

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
8.15  ParseException.java File Reference

Namespaces
  • namespace io

8.16  Parser.java File Reference

Namespaces
  • namespace io

8.17  ParserConstants.java File Reference

Namespaces
  • namespace io

8.18  ParserInterface.java File Reference

Defines the ParserInterface used to add objects to the current scene.

Namespaces
  • namespace io

8.18.1  Detailed Description

Defines the ParserInterface used to add objects to the current scene.
11 Sept. 2002

Author:
  Tyler Nielsen <nielsen@colorado.edu>

Date:
  2003/04/24 16:37:08

Revision:
  1.6

Definition in file ParserInterface.java.
8.19  ParserTokenManager.java File Reference

Namespaces

- namespace io

8.20  PickablePolygon.java File Reference

A drawable polygon coupled with a reference back to the SceneGraph.

Namespaces

- namespace disp

8.20.1  Detailed Description

A drawable polygon coupled with a reference back to the SceneGraph.

The polygon can be picked (with the mouse), and the reference can be followed to determine which _group_ the object belongs to.

19 Feb. 2003

Author:

Ryan D. Lewis <lewisrd@ucsub.colorado.edu>

Date:

2003/04/14 05:53:01

Revision:

1.13

Definition in file PickablePolygon.java.

8.21  PickingMouseListener.java File Reference

Used to ‘pick’ a polygon from the current scene.

Namespaces

- namespace win
8.21.1 Detailed Description

Used to 'pick' a polygon from the current scene.
26 Feb. 2003

Author:
Ryan D. Lewis <lewisrd@uicsub.colorado.edu>

Date:
2003/04/06 12:03:27

Revision:
1.6

Definition in file PickingMouseListener.java.

8.22 Polymesh.java File Reference

Implements a vertex-edge-face representation of 3D objects.

Namespaces

- namespace obj

8.22.1 Detailed Description

Implements a vertex-edge-face representation of 3D objects.
11 Jan. 2002

Author:
Ryan D. Lewis <lewisrd@uicsub.colorado.edu>

Date:
2003/04/14 06:09:05

Revision:
1.28

Definition in file Polymesh.java.

8.23 Render.java File Reference

Class to hold data needed to render a view.
Namespaces
  • namespace disp

8.23.1 Detailed Description
Class to hold data needed to render a view.
6 Oct. 2002

Author:
  Tyler Nielsen <nielsen@colorado.edu>

Date:
  2003/04/14 06:09:05

Revision:
  1.9

Definition in file Render.java.

8.24 Scene.java File Reference
Holds data relevant to the global scene, rather than individual windows.

Namespaces
  • namespace disp
  • namespace java.io
  • namespace javax.media.j3d

8.24.1 Detailed Description
Holds data relevant to the global scene, rather than individual windows.
14 Sept. 2002

Author:
  Tyler Nielsen <nielsen@colorado.edu>

Date:
  2003/05/04 21:16:47

Revision:
  1.61

Definition in file Scene.java.
8.25 SceneGraph.java File Reference

Implement a scene graph composed of Polymesh objects with Transformations applied to them.

Namespaces

- namespace `com.sun.j3d.utils.geometry`
- namespace `com.sun.j3d.utils.universe`
- namespace `java.awt.event`
- namespace `javax.vecmath`
- namespace `obj`

8.25.1 Detailed Description

Implement a scene graph composed of Polymesh objects with Transformations applied to them.

14 Feb. 2003 (yes, Valentine’s Day)

Author:
Ryan D. Lewis <lewisrd@ucsub.colorado.edu>

Date:
2003/05/05 00:26:04

Revision:
1.16

Definition in file `SceneGraph.java`.

8.26 SceneGraphNode.java File Reference

Implements the SceneGraphNode class, which makes up a SceneGraph.

Namespaces

- namespace `obj`
8.26.1 Detailed Description

Implements the SceneGraphNode class, which makes up a SceneGraph.
14 Feb. 2003

Author:
Ryan D. Lewis <lewisrd@ucsub.colorado.edu>

Date:
2003/04/30 19:45:12

Revision:
1.17

Definition in file SceneGraphNode.java.

8.27 SimpleCharStream.java File Reference

Namespaces

- namespace io

8.28 SpectreFileError.java File Reference

Defines a class of exceptions the parser can use to indicate various errors encountered when parsing a file.

Namespaces

- namespace io

8.28.1 Detailed Description

Defines a class of exceptions the parser can use to indicate various errors encountered when parsing a file.
5 Apr. 2003

Author:
Ryan D. Lewis <lewisrd@ucsub.colorado.edu>

Date:
2003/04/06 12:03:27
Revision:
1.2

Definition in file SpectreFileError.java.

8.29 SplashWindow.java File Reference

This class displayes a window while spectre is loading.

Namespaces

- namespace win

8.29.1 Detailed Description

This class displayes a window while spectre is loading.
11 Sept. 2002

Author:

Tyler Nielsen <nielsen@colorado.edu>

Date:

2003/04/25 00:58:12

Revision:

1.2

Definition in file SplashWindow.java.

8.30 StatusBar.java File Reference

Creates a status bar for ControlWindow which denotes current slice viewed.

Namespaces

- namespace win

8.30.1 Detailed Description

Creates a status bar for ControlWindow which denotes current slice viewed.
16 Sept. 2003
Author:
   Tyler Nielsen <nielsen@colorado.edu>

Date:
   2003/05/04 21:16:47

Revision:
   1.8

Definition in file StatusBar.java.

8.31  Token.java File Reference

Namespaces
   - namespace io

8.32  TokenMgrError.java File Reference

Namespaces
   - namespace io

8.33  Transform.java File Reference

Implements a Transformation that can be applied to SceneGraphNode.

Namespaces
   - namespace obj

8.33.1 Detailed Description

Implements a Transformation that can be applied to SceneGraphNode.
14 Feb. 2003

Author:
   Ryan D. Lewis <lewisrd@icsub.colorado.edu>

Date:
8.34 TransformFactory.java File Reference

Create Transforms for common purposes.

Namespaces

- namespace obj

8.34.1 Detailed Description

Create Transforms for common purposes.

14 Feb. 2003

Author:

Ryan D. Lewis <lewisrd@ucsub.colorado.edu>

Date:

2003/04/14 05:53:01

Version:

Revision:

1.8

Definition in file TransformFactory.java.
8.35 win.dox File Reference

Namespaces

- namespace win
Index

aboutmenu
  win::ControlWindow, 26
accept
  win::ExtensionFilter, 35
actionPerformed
  win::MenuHandler, 50
add
  disp::Scene, 131
  io::ParserInterface, 89
add_rmd
  io::Parser, 83
add_escapes
  io::ParseException, 74
addChild
  obj::SceneGraph, 142
  obj::SceneGraphNode, 149
addEscapes
  io::TokenMgrError, 178
addMenu
  win::ControlWindow, 22
addMenuItem
  win::ControlWindow, 22
addToolBar
  win::ControlWindow, 23
addToolBarButton
  win::ControlWindow, 23
addTransform
  obj::SceneGraphNode, 149
  obj::Transform, 181
addTransformNew
  obj::Transform, 182
adjustBeginLineColumn
  io::SimpleCharStream, 157
apply
  obj::Transform, 182
autoHighlight
  win::ObjDialog, 57
available
  io::SimpleCharStream, 162
backup
  io::SimpleCharStream, 157
beginColumn
  io::Token, 174
beginLine
  io::Token, 174
BeginToken
  io::SimpleCharStream, 158
bufcolumn
  io::SimpleCharStream, 162
buffer
  io::SimpleCharStream, 162
bufline
  io::SimpleCharStream, 163
bufpos
  io::SimpleCharStream, 163
bufsize
  io::SimpleCharStream, 163
calib
  win::OptionsDialog, 71
callback
  win::MenuHandler, 51
canvas3D
  win::J3DWindow, 44
CenterButton
  win::ObjDialog, 58
changed
  disp::Scene, 138
children
  obj::SceneGraphNode, 153
clearScene3D
  disp::Scene, 131
closeMenu
  win::ControlWindow, 26
COLOR
  io::ParserConstants, 85
color
  disp::PickablePolygon, 104
  obj::SceneGraphNode, 153
  win::StatusBar::ColorPane, 170
colorArgList
  io::Parser, 79
colorChooser
  disp::Scene, 138
colorChooserDialog
    disp::Scene, 138
colormenu
    win::ControlWindow, 26
ColorPane
    win::StatusBar::ColorPane, 170
colorPane
    win::StatusBar, 168
column
    io::SimpleCharStream, 163
com.sun.j3d.utils.behaviors.mouse, 14
com.sun.j3d.utils.geometry, 14
com.sun.j3d.utils.universe, 14
coords
    obj::Polymesh::HullSort, 116
coneTessellation
    obj::ObjFactory, 69
CONSTANT
    io::ParserConstants, 85
ControlWindow.java, 186
coords
    obj::Polymesh::Vertex, 124
cubeTessellation
    obj::ObjFactory, 69
curChar
    io::ParserTokenManager, 98
curlayer
    disp::Render, 127
curLexState
    io::ParserTokenManager, 98
currentObject
    disp::Scene, 138
currentToken
    io::ParseException, 75
curScene
    disp::Scene, 139
cylinderTessellation
    obj::ObjFactory, 69
dCount
    disp::Scene, 139
    win::ControlWindow, 26
    win::J3DWindow, 45
dCountMarkers
    disp::Scene, 139
    win::ControlWindow, 26
deplyStream
    io::ParserTokenManager, 99
DEFAULT
    io::ParserConstants, 86
defaultLayers
    disp::Scene, 139
defaultLexState
    io::ParserTokenManager, 99
degrees
    win::ObjDialog, 58
delete
    win::ControlWindow, 26
deleteChild
    obj::SceneGraph, 143
    obj::SceneGraphNode, 149
deleteSelectedObject
    disp::Scene, 131
description
    win::ExtensionFilter, 36
deHeight
    win::OptionsDialog, 71
dialog
    win::ObjDialog, 58
    win::OptionsDialog, 71
DIGIT
    io::ParserConstants, 86
dirty
    win::ControlWindow, 27
disable_tracing
    io::Parser, 79
disp, 14
disp.dox, 187
disp::Geometry, 36
getxMax, 38
getxMin, 38
getyMax, 38
getyMin, 38
getzMax, 38
getzMin, 38
inch2distX, 38
inch2distY, 39
inch2meter, 39
inch2pixX, 39
inch2pixY, 40

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
INDEX

layer2pixZ, 40
Printing, 41
setTranslation, 40
setzMax, 41
setzMin, 41
target, 41
xMax, 41
xMin, 42
yMax, 42
yMin, 42
zMax, 42
zMin, 42
disp::PickablePolygon, 102
disp::PickablePolygon
color, 104
getColor, 103
getPolygon, 103
getSceneGraphNode, 104
PickablePolygon, 103
pickReference, 104
polygon, 105
toString, 104
disp::Render, 125
curLayer, 127
getCurLayer, 126
moveDown, 126
moveUp, 126
setCurLayer, 126
disp::Scene, 127
add, 131
changed, 138
clearScene3D, 131
colorChooser, 138
colorChooserDialog, 138
currentObject, 138
curScene, 139
dCount, 139
dCountMarkers, 139
defaultLayers, 139
deleteSelectedObject, 131
drawCalib, 139
drawCalibrationObject, 131
drawCalibrationMarks, 132
drawGrid, 139
drawGridlines, 132
drawingColor, 139
drawScene, 132
drawScene3D, 133
drawSelectionMarkers, 133
filename, 140
getChanged, 133
getColor, 133
getCurrentObject, 134
getDirty, 134
getDirtyMarkers, 134
getDrawCalibrationMarks, 134
getDrawGridlines, 135
getFilename, 135
getLayers, 135
initScene, 135
layers, 140
Scene, 131
sceneGraph, 140
selectObject, 136
setChanged, 136
setColor, 136
setDirty, 136
setDirtyMarkers, 137
setDrawCalibrationMarks, 137
setDrawGridlines, 137
setFilename, 137
setLayers, 137
showColorChooserDialog, 138
toString, 138
Display

win::Display, 32

Display.java, 187

Done

io::SimpleCharStream, 158
dot

obj::Polymesh, 109
drawCalib
disp::Scene, 139
drawCalibrationMarks
disp::Scene, 132
drawFrame

win::ControlWindow, 24
drawGrid
disp::Scene, 139

win::OptionsDialog, 71
drawGridLines
disp::Scene, 132
drawingColor

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
INDEX

getDrawingColor
  io::Parser, 80
gGetEndColumn
  io::SimpleCharStream, 159
gGetEndLine
  io::SimpleCharStream, 159
gGetFileName
  disp::Scene, 135
gGetGeometryArray
  obj::SceneGraph, 143
gGetImage
  io::SimpleCharStream, 160
gGetIndex
  obj::PolyMesh::Vertex, 124
gGetLayers
  disp::Scene, 135
gGetLine
  io::SimpleCharStream, 160
gGetMatrix
  obj::Transform, 183
gGetMaxX
  obj::SceneGraphNode, 150
gGetMaxY
  obj::SceneGraphNode, 150
gGetMessage
  io::ParseException, 74
  io::TokenMgrError, 178
gGetMinX
  obj::SceneGraphNode, 151
gGetMinY
  obj::SceneGraphNode, 151
gGetNextToken
  io::Parser, 80
  io::ParserTokenManager, 91
gGetNumberOfPages
  win::ControlWindow, 24
gGetPageFormat
  win::ControlWindow, 24
gGetPolygon
  disp::PickablePolygon, 103
gGetPolyMesh
  obj::SceneGraphNode, 151
gGetPrintable
  win::ControlWindow, 24
gGetRoot
  obj::SceneGraph, 143
gGetSceneGraphNode
  disp::PickablePolygon, 104
GetSuffix
  io::SimpleCharStream, 160
gGetToken
  io::Parser, 80
gGetTransform
  obj::SceneGraphNode, 151
gGetXMax
  disp::Geometry, 38
gGetXMin
  disp::Geometry, 38
gGetYMax
  disp::Geometry, 38
gGetYMin
  disp::Geometry, 38
grid
  win::OptionsDialog, 71
hashCode
  obj::PolyMesh::Line, 118
height
  win::ObjDialog, 58
helpmenu
  win::ControlWindow, 27
hide
  win::ControlWindow, 25
  win::J3DWindow, 44
Hlabel
  win::ObjDialog, 58
Hpanel
  win::ObjDialog, 58
Htext
  win::ObjDialog, 58
HullSort
  obj::PolyMesh::HullSort, 115
image
  io::Token, 175
importMenu
  win::ControlWindow, 27
inBuf
io::ParserTokenManager, 89
io::ParserTokenManager
curChar, 98
curLexState, 98
debugStream, 99
defaultLexState, 99
getNextToken, 91
input_stream, 99
jjAddStates, 92
jjCheckNAdd, 92
jjCheckNAddStates, 92
jjCheckNAddTwoStates, 92
jjFillToken, 93
jjMatchedKind, 99
jjMatchedPos, 99
jjMoveNfa_0, 93
jjMoveStringLiteralDfa0_0, 93
jjMoveStringLiteralDfa1_0, 93
jjMoveStringLiteralDfa2_0, 94
jjMoveStringLiteralDfa3_0, 94
jjMoveStringLiteralDfa4_0, 94
jjMoveStringLiteralDfa5_0, 95
jjMoveStringLiteralDfa6_0, 95
jjMoveStringLiteralDfa7_0, 95
jjMoveStringLiteralDfa8_0, 95
jjMoveStringLiteralDfa9_0, 96
jNewStateCnt, 99
jNextStates, 100
jRound, 100
jRounds, 100
jjStartNfa_0, 96
jjStartNfaWithStates_0, 96
jjStateSet, 100
jjStopAtPos, 96
jjStopStringLiteralDfa_0, 97
jstrLiteralImages, 100
jToSkip, 101
jToToken, 101
lexStateNames, 101
ParserTokenManager, 91
ReInit, 97
ReInitRounds, 97
setDebugStream, 98
SwitchTo, 98
io::SimpleCharStream, 154
io::SimpleCharStream
adjustBeginLineColumn, 157
available, 162
backup, 157
BeginToken, 158
bufcolumn, 162
buffer, 162
bufline, 163
bufpos, 163
bufsize, 163
column, 163
Done, 158
ExpandBuff, 158
FillBuff, 158
getBeginColumn, 158
getBeginLine, 159
column, 159
column, 159
column, 159
column, 159
column, 159
Done, 158
ExpandBuff, 158
FillBuff, 158
getBeginColumn, 158
getBeginLine, 159
column, 159
maxNextCharInd, 164
prevCharIsCR, 164
prevCharIsLF, 164
readChar, 160
ReInit, 161
SimpleCharStream, 156, 157
statFlag, 164
tokenBegin, 165
UpdateLineColumn, 162
io::SpectreFileStream, 165
io::SpectreFileStream
SpectreFileStream, 166
io::Token, 172
beginColumn, 174
beginLine, 174
dColon, 175
dLine, 175
image, 175
kind, 175
newToken, 174

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
INDEX

jjMoveStringLiteralDfa9_0
  io::ParserTokenManager, 96
jjnewStateCnt
  io::ParserTokenManager, 99
jjnextStates
  io::ParserTokenManager, 100
jjround
  io::ParserTokenManager, 100
jjrounds
  io::ParserTokenManager, 100
jjStartNfa_0
  io::ParserTokenManager, 96
jjStartNfaWithStates_0
  io::ParserTokenManager, 96
jjstateSet
  io::ParserTokenManager, 100
jjStopAtPos
  io::ParserTokenManager, 96
jjStopStringLiteralDfa_0
  io::ParserTokenManager, 97
jjstrLiteralImages
  io::ParserTokenManager, 100
jjtoSkip
  io::ParserTokenManager, 101
jjtoToken
  io::ParserTokenManager, 101

kind
  io::Token, 175
knownGroups
  obj::SceneGraph, 145

layer2pixZ
  disp::Geometry, 40
layerPane
  win::StatusBar, 168
layers
  disp::Scene, 140
length
  win::ObjDialog, 59
LEXICAL_ERROR
  io::TokenMgrError, 179
LexicalError
  io::TokenMgrError, 178
lexStateNames
  io::ParserTokenManager, 101

license
  win::ControlWindow, 27
Line
  obj::PolyMesh::Line, 118
line
  io::SimpleCharStream, 164
Label
  win::ObjDialog, 59
loadScene
  win::MenuHandler, 50
LOOP_DETECTED
  io::TokenMgrError, 179
Lpanel
  win::ObjDialog, 59
Ltext
  win::ObjDialog, 59

main
  win::MainWindow, 46
mainpage.dox, 189
MainWindow.java, 189
majorVersion
  win::MainWindow, 47
manual
  win::ControlWindow, 28
maxNextCharInd
  io::SimpleCharStream, 164
menubar
  win::ControlWindow, 28
MenuHandler
  win::MenuHandler, 49, 50
MenuHandler.java, 190
minorVersion
  win::MainWindow, 47
mouseClicked
  win::PickingMouseListener, 106
moveDown
  disp::Render, 126
moveUp
  disp::Render, 126
new2dview
  win::ControlWindow, 28
new3D
  win::Display, 32

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
new3dview
win::ControlWindow, 28
newCone
obj::ObjFactory, 68
newone
win::ControlWindow, 28
newCube
obj::ObjFactory, 68
newcube
win::ControlWindow, 28
newCylinder
obj::ObjFactory, 69
newcylinder
win::ControlWindow, 28
newmenu
win::ControlWindow, 28
newRotateX
obj::TransformFactory, 184
newRotateY
obj::TransformFactory, 184
newRotateZ
obj::TransformFactory, 185
newScale
obj::TransformFactory, 185
newSlice
win::Display, 32
newSphere
obj::ObjFactory, 69
newsphere
win::ControlWindow, 28
newToken
io::Token, 174
newTranslate
obj::TransformFactory, 186
next
io::Token, 176
nextmenu
win::ControlWindow, 28
NODE
io::ParserConstants, 86
node
io::Parser, 81
nodeArgList
io::Parser, 81
normal
obj::PolyMesh::Plane, 120
numSlices
win::OptionsDialog, 71
obj, 16
obj.dox, 191
obj::ObjFactory, 67
obj::ObjFactory
coneTessellation, 69
cubeTessellation, 69
cylinderTessellation, 69
newCone, 68
newCube, 68
newCylinder, 69
newSphere, 69
sphereTessellation, 69
obj::PolyMesh, 107
dot, 109
edgeify, 110
eList, 113
fList, 113
intersectLinePlane, 110
PolyMesh, 109
slice, 111
toString, 111
transformLine, 111
vertAdd, 112
vertMult, 112
verts, 113
vertSub, 113
xMax, 113
xMin, 114
yMax, 114
yMin, 114
obj::PolyMesh::HullSort, 114
obj::PolyMesh::HullSort
compare, 116
HullSort, 115
xCenter, 116
tyCenter, 116
obj::PolyMesh::Line, 117
equals, 118
hashCode, 118
Line, 118
toString, 118
v1, 119
v2, 119
obj::Polymesh::Plane, 119
  normal, 120
  Plane, 120
  point, 120
obj::Polymesh::Tri, 121
  toString, 122
  Tri, 122
  v, 122
obj::Polymesh::Vertex, 122
  coords, 124
  getIndex, 124
  index, 125
  setIndex, 124
  toString, 124
  Vertex, 123
obj::SceneGraph, 140
obj::SceneGraph
  addChild, 142
  deleteChild, 143
  getGeometryArray, 143
  getRoot, 143
  J3DObj, 143
  J3DScene, 144
  knownGroups, 145
  root, 145
  SceneGraph, 142
  slice, 144
  toString, 145
obj::SceneGraphNode, 145
obj::SceneGraphNode
  addChild, 149
  addTransform, 149
  children, 153
  color, 153
  deleteChild, 149
  fromOriginTransform, 150
  getChildren, 150
  getColor, 150
  getMaxX, 150
  getMaxY, 150
  getMinX, 151
  getMinY, 151
  getPolymesh, 151
  getTransform, 151
  poly, 153
  SceneGraphNode, 148

setColor, 151
setPolymesh, 152
slice, 152
toOriginTransform, 152
toString, 152
trans, 153
updateBounds, 153
xMax, 154
xMin, 154
yMax, 154
yMin, 154
obj::Transform, 180
  addTransform, 181
  addTransformNew, 182
  apply, 182
  getMatrix, 183
  tMatrix, 183
  toString, 183
  Transform, 181
obj::TransformFactory, 183
obj::TransformFactory
  newRotateX, 184
  newRotateY, 184
  newRotateZ, 185
  newScale, 185
  newTranslate, 186
ObjDialog.java, 191
ObjFactory.java, 191
objList Reference
  win::ControlWindow, 29
  win::PickingMouseListener, 107
Objx
  win::ObjDialog, 59
ObjXlabel
  win::ObjDialog, 59
ObjXpanel
  win::ObjDialog, 60
ObjXtext
  win::ObjDialog, 60
Objy
  win::ObjDialog, 60
ObjYlabel
  win::ObjDialog, 60
ObjYpanel
  win::ObjDialog, 60
ObjYtext
    win::ObjDialog, 60
Objz
    win::ObjDialog, 61
ObjZlabel
    win::ObjDialog, 61
ObjZpanel
    win::ObjDialog, 61
ObjZtext
    win::ObjDialog, 61
Olabel
    win::ObjDialog, 61
openmenu
    win::ControlWindow, 29
OptionsDialog.java, 192
optionsmenu
    win::ControlWindow, 29
OriginButton
    win::ObjDialog, 61
OriginCenter
    win::ObjDialog, 61
paintComponent
    win::StatusBar::ColorPane, 170
panelFont
    win::StatusBar::StatusPane, 172
parseCurrentFile
    io::Parser, 81
ParseException
    io::ParseException, 73, 74
ParseException.java, 193
Parser
    io::Parser, 78, 79
parser
    win::MenuHandler, 52
Parser.java, 193
ParserConstants.java, 193
ParserInterface.java, 193
ParserTokenManager
    io::ParserTokenManager, 91
ParserTokenManager.java, 194
PickingMouseListener
    win::PickingMouseListener, 106
PickingMouseListener.java, 194
pickReference
    disp::PickablePolygon, 104
Plane
    obj::Polymesh::Plane, 120
point
    obj::Polymesh::Plane, 120
poly
    obj::SceneGraphNode, 153
polygon
    disp::PickablePolygon, 105
POLYMESH
    io::ParserConstants, 86
Polymesh
    obj::Polymesh, 109
Polymesh.java, 195
polymeshArgList
    io::Parser, 81
prevCharIsCR
    io::SimpleCharStream, 164
prevCharIsLF
    io::SimpleCharStream, 164
prevmenu
    win::ControlWindow, 29
print
    win::ControlWindow, 25
printCalib
    win::OptionsDialog, 71
Printing
    disp::Geometry, 41
printmenu
    win::ControlWindow, 29
radius
    win::ObjDialog, 62
readChar
    io::SimpleCharStream, 160
readInput
    win::ObjDialog, 56
ReInit
    io::Parser, 82
    io::ParserTokenManager, 97
    io::SimpleCharStream, 161
<table>
<thead>
<tr>
<th>Function/Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReInitRounds</td>
<td>io::ParserTokenManager, 97</td>
</tr>
<tr>
<td>remove2D</td>
<td>win::Display, 32</td>
</tr>
<tr>
<td>remove3D</td>
<td>win::Display, 33</td>
</tr>
<tr>
<td>Render.java</td>
<td>195</td>
</tr>
<tr>
<td>resetDefaults</td>
<td>win::ObjDialog, 56</td>
</tr>
<tr>
<td>Rlabel</td>
<td>win::ObjDialog, 62</td>
</tr>
<tr>
<td>root</td>
<td>obj::SceneGraph, 145</td>
</tr>
<tr>
<td>rotateX</td>
<td>win::ControlWindow, 29</td>
</tr>
<tr>
<td>rotateY</td>
<td>win::ControlWindow, 29</td>
</tr>
<tr>
<td>rotateZ</td>
<td>win::ControlWindow, 29</td>
</tr>
<tr>
<td>Rpanel</td>
<td>win::ObjDialog, 62</td>
</tr>
<tr>
<td>Rtext</td>
<td>win::ObjDialog, 62</td>
</tr>
<tr>
<td>runAsApplet</td>
<td>win::MainWindow, 47</td>
</tr>
<tr>
<td>savasmenu</td>
<td>win::ControlWindow, 29</td>
</tr>
<tr>
<td>savemenu</td>
<td>win::ControlWindow, 30</td>
</tr>
<tr>
<td>saveScene</td>
<td>win::MenuHandler, 50</td>
</tr>
<tr>
<td>saveScenePrompt</td>
<td>win::MenuHandler, 51</td>
</tr>
<tr>
<td>scale</td>
<td>win::ControlWindow, 30</td>
</tr>
<tr>
<td>Scene</td>
<td>disp::Scene, 131</td>
</tr>
<tr>
<td>Scene.java</td>
<td>196</td>
</tr>
<tr>
<td>SceneGraph</td>
<td>obj::SceneGraph, 142</td>
</tr>
<tr>
<td>sceneGraph</td>
<td>disp::Scene, 140</td>
</tr>
<tr>
<td>SceneGraph.java</td>
<td>197</td>
</tr>
<tr>
<td>SceneGraphNode</td>
<td>obj::SceneGraphNode, 148</td>
</tr>
<tr>
<td>SceneGraphNode.java</td>
<td>197</td>
</tr>
<tr>
<td>sceneRot</td>
<td>win::J3DWindow, 45</td>
</tr>
<tr>
<td>SCI</td>
<td>io::ParserConstants, 87</td>
</tr>
<tr>
<td>Sclx</td>
<td>win::ObjDialog, 62</td>
</tr>
<tr>
<td>Scdx</td>
<td>win::ObjDialog, 62</td>
</tr>
<tr>
<td>Scproxy</td>
<td>win::ObjDialog, 62</td>
</tr>
<tr>
<td>Scy</td>
<td>win::ObjDialog, 63</td>
</tr>
<tr>
<td>ScyPanel</td>
<td>win::ObjDialog, 63</td>
</tr>
<tr>
<td>ScyText</td>
<td>win::ObjDialog, 63</td>
</tr>
<tr>
<td>Scz</td>
<td>win::ObjDialog, 63</td>
</tr>
<tr>
<td>SczPanel</td>
<td>win::ObjDialog, 64</td>
</tr>
<tr>
<td>SczText</td>
<td>win::ObjDialog, 64</td>
</tr>
<tr>
<td>selectObject</td>
<td>disp::Scene, 136</td>
</tr>
<tr>
<td>setChanged</td>
<td>disp::Scene, 136</td>
</tr>
<tr>
<td>setColor</td>
<td>disp::Scene, 136</td>
</tr>
<tr>
<td>setColor</td>
<td>io::ParserInterface, 89</td>
</tr>
<tr>
<td>setUser</td>
<td>obj::SceneGraphNode, 151</td>
</tr>
<tr>
<td>win::StatusBar::ColorPane, 170</td>
<td></td>
</tr>
<tr>
<td>setColor</td>
<td>win::ControlWindow, 30</td>
</tr>
<tr>
<td>setCurLayer</td>
<td>win::StatusBar, 168</td>
</tr>
<tr>
<td>setDebugStream</td>
<td>disp::Render, 126</td>
</tr>
</tbody>
</table>
INDEX

okenBegin
  io::SimpleCharStream, 165
okenImage
  io::ParseException, 75
  io::ParserConstants, 87
TokenMgrError
  io::TokenMgrError, 177, 178
TokenMgrError.java, 200
toOriginTransform
  obj::SceneGraphNode, 152
toString
  disp::PickablePolygon, 104
  disp::Scene, 138
  io::Token, 174
  obj::Polymesh, 111
  obj::Polymesh::Line, 118
  obj::Polymesh::Tri, 122
  obj::Polymesh::Vertex, 124
  obj::SceneGraph, 145
  obj::SceneGraphNode, 152
  obj::Transform, 183
trans
  obj::SceneGraphNode, 153
TRANSFORM
  io::ParserConstants, 88
Transform
  obj::Transform, 181
Transform.java, 200
transformArgList
  io::Parser, 83
TransformFactory.java, 201
transformLine
  obj::Polymesh, 111
translate
  win::ControlWindow, 30
Tranx
  win::ObjDialog, 65
TranXlabel
  win::ObjDialog, 65
TranXpanel
  win::ObjDialog, 65
TranXtext
  win::ObjDialog, 65
Tranx
  win::ObjDialog, 65
TranYlabel
  win::ObjDialog, 65
TranYpanel
  win::ObjDialog, 65
TranYtext
  win::ObjDialog, 66
Tranz
  win::ObjDialog, 66
TranZlabel
  win::ObjDialog, 66
TranZpanel
  win::ObjDialog, 66
TranZtext
  win::ObjDialog, 66
Tri
  obj::Polymesh::Tri, 122
updateBounds
  obj::SceneGraphNode, 153
UpdateLineColumn
  io::SimpleCharStream, 162
updateWindow
  win::ControlWindow, 26
  win::Display, 33
  win::J3DWindow, 44
  obj::Polymesh::Tri, 122
v
  obj::Polymesh::Line, 119
v1
  obj::Polymesh::Line, 119
v2
  obj::Polymesh::Line, 119
vertAdd
  obj::Polymesh, 112
Vertex
  obj::Polymesh::Vertex, 123
VERT LIST
  io::ParserConstants, 88
vertMult
  obj::Polymesh, 112
verts
  obj::Polymesh, 113
vertSub
  obj::Polymesh, 113
view

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen
runAsApplet, 47
window, 47
win::MenuHandler, 47
win::MenuHandler
actionPerformed, 50
callback, 51
classes, 51
loadScene, 50
MenuHandler, 49, 50
parser, 52
saveScene, 50
saveScenePrompt, 51
showDialog, 51
win::ObjDialog, 52
win::ObjDialog
autoHighlight, 57
CenterButton, 58
degrees, 58
dialog, 58
height, 58
Hlabel, 58
Hpanel, 58
Htext, 58
length, 59
Llabel, 59
Lpanel, 59
Ltext, 59
Objx, 59
ObjXLabel, 59
ObjXPanel, 60
ObjXText, 60
Objy, 60
ObjYLabel, 60
ObjYPanel, 60
ObjYText, 60
Objz, 61
ObjZLabel, 61
ObjZPanel, 61
ObjZText, 61
Olabel, 61
OriginButton, 61
OriginCenter, 61
radius, 62
readInput, 56
resetDefaults, 56
Rlabel, 62
Rpanel, 62
RText, 62
ScLx, 62
ScLxLabel, 62
ScLxPanel, 62
ScLxText, 63
ScLy, 63
ScLxYLabel, 63
ScLxYPanel, 63
ScLxYText, 63
ScLz, 63
ScLzLabel, 64
ScLzPanel, 64
ScLzText, 64
setPanels, 56
showDialog, 57
Spanel, 64
Stext, 64
Tranx, 65
TranXLabel, 65
TranXPanel, 65
TranXText, 65
Trany, 65
TranYLabel, 65
TranYPannel, 65
TranYText, 66
Tranz, 66
TranZLabel, 66
TranZPanel, 66
TranZText, 66
width, 66
Wlabel, 66
Wpanel, 67
Wtext, 67
win::OptionsDialog, 70
win::OptionsDialog
calib, 71
devHeight, 71
dialog, 71
drawGrid, 71
grid, 71
numSlices, 71
printCalib, 71
showDialog, 70
win::PickingMouseListener, 105
INDEX

yCenter
  obj::Polymesh::HullSort, 116
yMax
  disp::Geometry, 42
  obj::Polymesh, 114
  obj::SceneGraphNode, 154
yMin
  disp::Geometry, 42
  obj::Polymesh, 114
  obj::SceneGraphNode, 154
zMax
  disp::Geometry, 42
zMin
  disp::Geometry, 42

Generated on Mon May 5 20:55:00 2003 for SPECTRE by Doxygen