

ThreeSpace  
DESIGN DOCUMENT

Adam Chodorowski & John Nilsson

June 28, 2001

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	3
1.3	Definitions . . . . .	3
1.4	References . . . . .	3
<b>2</b>	<b>Module description</b>	<b>5</b>
2.1	Scene module . . . . .	5
2.1.1	Scene component . . . . .	5
2.1.2	Shape component . . . . .	5
2.1.3	Geometry component . . . . .	5
2.2	User Interface module . . . . .	6
2.2.1	ViewPanel component . . . . .	6
2.2.2	PropertyPanel component . . . . .	6
2.2.3	ToolbarPanel component . . . . .	6
2.2.4	Menubar component . . . . .	6
2.3	Plugin module . . . . .	7
2.3.1	PluginManager component . . . . .	7
2.3.2	Plugin component . . . . .	7
<b>3</b>	<b>Use cases</b>	<b>8</b>
3.1	Overview . . . . .	8

# Chapter 1

## Introduction

### 1.1 Purpose

The purpose of this software design specification is to provide an design view of ThreeSpace. It shall provide sufficient information on structural organisation and component composition. It shall not specify implementation-specific information. This document is based on the standard IEEE 1016-1998.

### 1.2 Scope

ThreeSpace is a 3D drawing program. Its design is driven to allow easy scaling by allowing external plugins access its 3D scene and objects through well-defined interfaces.

### 1.3 Definitions

- **Object** - An object is a collection of polygons that is handled as a single entity. An object has a shape in 3D.
- **Shape** - The shape of an object is the way an object looks. The shape of an object is specified by its polygons and their orientation.
- **Scene** - The scene is the 3D space in which all objects are created. It encompasses all currently visible and not visible objects that have been added to the 3D space by the user.

Please see the Requirements Document for the full list of definitions.

### 1.4 References

1. IEEE Std 1016-1998 “IEEE Recommended Practice for Software Design Descriptions.”
2. Java 2 SDK API Documentation, Sun Microsystems  
<http://java.sun.com/j2se/1.3/docs/api/>
3. Java3D API Documentation, Sun Microsystems  
<http://java.sun.com/products/java-media/3D/releases.html>

4. Java3D API Tutorial, Sun Microsystems  
<http://java.sun.com/products/java-media/3D/collateral/>
5. "Debugging Java", Will David Mitchell

# Chapter 2

## Module description

### 2.1 Scene module

#### 2.1.1 Scene component

The Scene component handles:

- The adding, deleting, selecting, hiding, showing of objects visible in the Scene.
- The properties of the 3D-view, its background color and the visibility of its axis.
- The rendering mode (ie. Shaded, Shaded without Gouraud shading, Wireframe, Dotted)
- The receiving of Mouse & Keyboard events and passing it along to registered listeners.

#### Dependency

The Scene component depends on the Shape component.

#### 2.1.2 Shape component

The Shape component handles:

- Positioning, rotation, color and modification of visual properties.
- The filter lists for Colorfilters, Structurefilters, Rotationfilters and Positionfilters.

It does not change the structure of the scene object but does provide a reference to the Geometry.

#### Dependency

The Shape component depends on the Geometry component.

#### 2.1.3 Geometry component

A component which handles the structural composition of the geometry of an object. It handles the vertices that make up the scene object and their attributes, ie. color and position. It calculates the normals for the polygons, which is used for shading, and provides the Java3D Geometry object which is later inserted to the Java3D components. Generally it handles:

- The getting and setting of Vertex data
- The getting and setting of Vertex attributes

## Dependencies

The component does not depend on any other component.

## 2.2 User Interface module

### 2.2.1 ViewPanel component

A component which adds the Java3D view component received from the Scene component.

### 2.2.2 PropertyPanel component

A component which handles the layout of the *Property Panel* in Camera state and Object state. The Camera state provide controls for changing:

- Camera rotation, position
- Adding, removing and selecting Camera bookmarks

The Object state provide controls for changing:

- Object name, position, rotation and color
- Filter lists for color, structure, rotation and position

### 2.2.3 ToolbarPanel component

A component which provide buttons for the most important actions in the program. There exist buttons for the following actions:

1. New scene
2. Load scene
3. Save scene
4. Save scene as
5. Switch to Camera State
6. Switch to Object State
7. Switch to rendering mode 1, Gouraud Shading
8. Switch to rendering mode 2, Flat Shading
9. Switch to rendering mode 3, Wireframe mode
10. Switch to rendering mode 4, Point mode

### 2.2.4 Menubar component

A component which provides a menu for:

- Loading, Saving, and Exporting the current View
- Creating, Loading, Saving, Hiding and Showing objects
- Adding filters of Color,Structure,Position and Rotation type
- Setting background color and adding/removing the axis

## 2.3 Plugin module

### 2.3.1 PluginManager component

A component which handles the loading and creation of Plugin-objects. It provides lists which the application can use to populate the menu and some parts of the *Property Panel*. The Plugin-Manager loads classes from directories and recognises their implemented interfaces.

### 2.3.2 Plugin component

A component with a clear interface so that third-party developers can create plugins. The component shall provide dialogs for user interaction and should modify objects and object geometry according to user selection. A plugin can also create an entirely new object based on dialog-interaction from the user.

#### Interfaces

The Plugin interface provides:

- Getting the name of the plugin
- Getting the version of the plugin
- Getting a brief description of the plugin

#### Interfaces that extend the Plugin interface

The ShapeCreator provides an interface to:

- Get a panel in which the user can enter properties for the ShapeCreator.
- A method which creates an actual shape which later becomes the SceneObject.
- A method which returns the preferred name of the object created.

The Filter provides an interface to:

- Get a panel with filter properties.
- Apply the filter to a specified shape.

The SceneLoader provides an interface to:

- Loading the entire scene from a given filename with a specific protocol.

The SceneSaver provides an interface to:

- Saving the entire scene to a given filename with a specific protocol.

The ObjectLoader provides an interface to:

- Loading Shapes from a given filename with a specific protocol.

The ObjectSaver provides an interface to:

- Saving Shapes to a given filename with a specific protocol.

The ViewExporter provides an interface to:

- Exporting the current view to a given filename with a specific protocol

## Chapter 3

# Use cases

### 3.1 Overview

This is a general overview of the actions available to the user.



