

ThreeSpace
USERS MANUAL

Adam Chodorowski & John Nilsson

June 28, 2001

Contents

1	Introduction	3
2	Requirements	4
3	Using	5
3.1	Navigating in the scene	5
3.2	Creating objects	5
3.3	Selecting objects	6
3.4	Moving and rotating objects	6
3.5	Applying filters	6

Chapter 1

Introduction

ThreeSpace is a 3D drawing program, or modeller if you will, primarily intended for visualizing and playing with mathematical functions. It allows you to superimpose these or otherwise alter objects in 3D using them.

The main idea behind ThreeSpace is that of *filters*. These are similar to filters and effects that are common in 2D graphics program, but here extended to work in 3 dimensions. Basically, each object you create in ThreeSpace can be assigned layers of *filters* that modify it's appearance in some way. There is no limitation on how many filters you can apply to an object, as long as there's enough memory. Also, you can at any time remove *filters* and return to the state the object was in before having applied them. In this way, *filters* are a bit similar to the concept of layers in some other programs.

When it comes to mathematical functions, you can simply enter them as normal free text and they will be parsed and used in the way specified. ThreeSpace uses JEL (the GNU Java Expression Library) to parse and compile the functions into Java bytecode, which makes them relatively fast to use even on large objects. Mathematical functions can be used to create objects from scratch or as parameters to different filters, where they serve as a description where and how the filter should be applied.

This manual assumes that you have some previous knowledge of 3D programs and computers in general and is not intended for novices. This might change in the future, but for now we recommend you to read a tutorial on 3D graphics if you feel you need it. Also, it might be useful to read the *Requirements Document* for a full list of words used in ThreeSpace and some explanations.

Chapter 2

Requirements

ThreeSpace requires the following things to run, most of which are not included:

- Java 2 Standard Edition 1.2 or higher.
- Java3D 1.1 or higher.
- JEL 0.9.7 or higher (included with the package).

Chapter 3

Using

When you first start ThreeSpace, you will be presented with an empty scene. The view just consists of a black background and the X (red), Y (green) and Z (yellow) axis that help you orientate yourself in the scene.

3.1 Navigating in the scene

The camera is always centered around a specific point in space that it "looks at", and positioning is always in reference to this point. When you start with an empty scene, this point is most likely the origin (ie. center of the scene). After you have created some objects, you can center the camera around one of them by double-clicking on it with the left mouse button when in *camera state*.

There are two ways of changing the camera's view in ThreeSpace: by using the mouse or the *property panel*. Using the mouse is in most cases the fastest and most intuitive, while the *property panel* provides exact positioning.

Using the mouse, all three mouse-buttons are used. If you only have two buttons (or heaven forbid, just one) you're unfortunately out of luck unless your OS provides some means of emulating the missing buttons. Anyway, you can do the following things by holding down the specified mouse button and moving the mouse:

- **[Left button]** Rotates you around the X- and Y-axis.
- **[Middle button]** Rotates you around the Z-axis.
- **[Right button]** Moves the camera closer or further away from the center-point.

Using the *property panel*, you can set the exact coordinates of the center-point, the camera's distance to it and the camera's location using the appropriate textfields, sliders or buttons.

3.2 Creating objects

Now, let's create some content for the scene. ThreeSpace can create several different shapes from scratch, and you access all of them through the **Object -> New** submenu. Simply select the menu-item with the shape you want, and the *create dialog* will open.

The top of the *create dialog* contains some information about the plugins that creates the shape you've selected. Below that are some plugin-specific settings that you can modify. Most of them should be easy to understand (like "length" and "width"), but some may require a bit of explanation. Unfortunately, you won't find that information in this manual, since it is totally plugin-specific. You should consult the manual of the plugin for more information.

After you've selected the parameters you want (or just leaving the default), you can click on the "create" button and an object will be created at the default location (the origin) and default colors. Clicking "cancel" simply cancels the dialog, and no object is created.

ThreeSpace automatically switches to *object state* and selects the object that was recently created.

3.3 Selecting objects

To select an object for some editing (moving, rotating, applying filters, etc.) you simply double-click on it whilst in *object state*. This can however fail sometimes, because objects overlap, and therefore you can alternatively use the space-bar to switch between the objects in the scene.

3.4 Moving and rotating objects

To move or rotate an object, you first have to switch to *object state* and select the object. In the same way as in *camera state*, you can now rotate and position the object by either using the *property panel* or the mouse.

All three mouse-buttons are used here too, and by holding one of them down and moving the mouse the following happens:

- **[Left button]** Rotates the object around its axis.
- **[Middle button]** Moves the object in the direction of the camera's *look vector*.
- **[Right button]** Moves the object in the plane the is perpendicular to the camera's *look vector*.

Just try it, it's the easiest way to learn. :-)

Using the *property panel*, you can set the objects exact position and rotation using the appropriate texfields, sliders and buttons.

3.5 Applying filters

Currently ThreeSpace supports four types of filters, that modify the object in different ways:

- Structure filters, which modify the structure (ie. vertices and polygons) of an object.
- Color filters, which modify how an object is colored.
- Position filters, which modify an objects position or specify where an object can be moved by the user.
- Rotation filters, which modify an objects rotation around it's axis or specify how the object can be rotated.

To add some filters to an object, you first have to switch to *object state* and select the specific object. Then you can either select which filter you want to apply directly from the **Object** menu (ie. the **Object -> Structure** , **Object -> Color** , **Object -> Position** and **Object -> Rotation** sub-menus). This will automatically add the selected filter to the object and open the *filter dialog*. You can also open the *filter dialog* by clicking on one of the "filters" buttons in the *property panel* (choosing the one that corresponds to the type of filter you wish to add).

In both cases, the *filter dialog* looks the same. To the left there is a list of filters that have been added to the object. To the right of the list you have four buttons with which you can re-arrange the ordering of the filters (which affects the way the result will be after applying them). Below the list you have two buttons which you can use to add additional filters or remove already added ones.

When you select a filter, the right part of the *filter dialog* will show filter-specific settings that you can modify (if any). Most of them should be self-explanatory, but some might require a bit of explanation. Since they are filter-specific they are not discussed in this manual, and you should read the manual of the filter for more information. In the future, they might be included as appendices to this manual.

When you're done, simply click on the "apply" button and the filters will be applied and the display updated (if necessary). Clicking on the "cancel" button removes any filters you have added, and returns the state of the object to the one it had before the *filter dialog* was opened.