3D Graphics Tools

Developer

Reference

Series

September 1995

How to Use This Manual

This manual describes the functions and operating procedure of 3D Graphics Tool (DTL-S220A).

The manual consists of:

PART 1. INTRODUCTION

Describes the system needed to run 3D Graphics Tool, installation procedure, and outline of the entire tool.

PART 2. TUTORIAL

Explains the use of 3D Graphics Tool using examples and exercises.

PART 3. REFERENCE

Describes all the functions of 3D Graphics Tool.

APPENDIX. FILE FORMAT

Explains file format used for 3D Graphics Tool.

Tool Versions Covered by this manual:

This manual is intended for the following versions, and later:

ABOARD.EXE	1.00
ANIMATION.EXE	1.1.5
LIBDATA.DLL	1.1.4
LIB3D.DLL	1.1.1
TODEXP.DLL	1.0.1
BETWEEN.DLL	1.0.1
BMP2TIM.EXE	1.0
DXF2RSD.EXE	2.7
DXF2RSDW.EXE	1.10
MEDITOR.EXE	1.70
MIMEFILT.EXE	1.2
MIMESORT.EXE	1.0
MKTOD.EXE	1.3
PICT2TIM.EXE	3.1
RGB2TIM.EXE	2.0
RSD2DXF.EXE	1.10
RSDCAT.EXE	1.03
RSDFORM.EXE	1.8
RSDLINK.EXE	3.6

TIMEXP.8BE	1.2
TIMFMT.8BI	1.2
TIMPOS.EXE	1.0
TIMUTIL.EXE	1.35
TMDSORT	1.1
TMD2PMD.EXE	1.12
TMDSORT.EXE	1.1

CONTENTS

3D EDITION

How to Use this Manual

Contents

PART 1

INTRODUCTION

CHAPTER 1

SYSTEM CONFIGURATION

This chapter describes system configuration needed to use 3D Graphics Tool.

To use 3D Graphics Tool, the following hardware and software are needed.

1 Hardware

IBM PC/AT or compatible

The machine must satisfy the following conditions:

Hard disk
CPU equivalent to, or greater than, 486DX2 66-MHz
Free memory space of 4 M-bytes or more
SVGA monitor

Graphics Artist Board DTL-H201A

Mounted in the expansion slot in PC/AT. A color monitor suitable for NTSC is connected to it. It is used by the material editor and convertor.

Color monitor for PC/AT

3D Graphics Tool performs major operations on a color monitor for PC/AT.

Video Monitor

Used to display output from the Graphics Artist Board.

Mouse

The tool is mostly manipulated using the mouse.

2 Software

MS-DOS

Use Version 5.0 or later.

Windows3.1

3D Graphics Tool runs on Japanese Language Windows3.1 available from Microsoft Corporation.

CHAPTER 2

INSTALLATION

This chapter describes how to install 3D Graphics Tool.

1 Installation Floppy Disk

- Start Windows.
- Terminate all the programs, excepting Program Manger and File Manager.
- Insert “Setup Disk 1” into the floppy disk drive.
- Open drive A using File Manager.

Figure 2-1-1 File Manager

- Double-click “setup.exe” to start Installer.

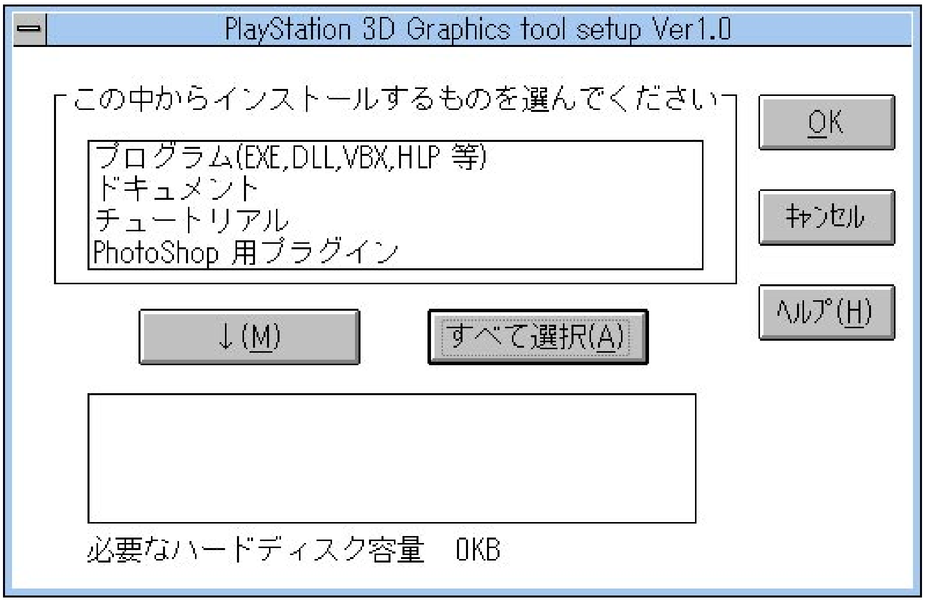


Figure 2-1-2 Installer Start Screen

- You can select “Program” (EXE.DLL, VBX.HLP, etc)”, “Document”, “Tutorial”, or “Plug-in for PhotoShop”.

Select a necessary item among them, and press the “Ø(M)” button. The item list appears on the lower field. When you have selected all the necessary items, press the “OK” button.

A dialogue box pops up, asking you to set the drive and directory where the programs are to be installed.

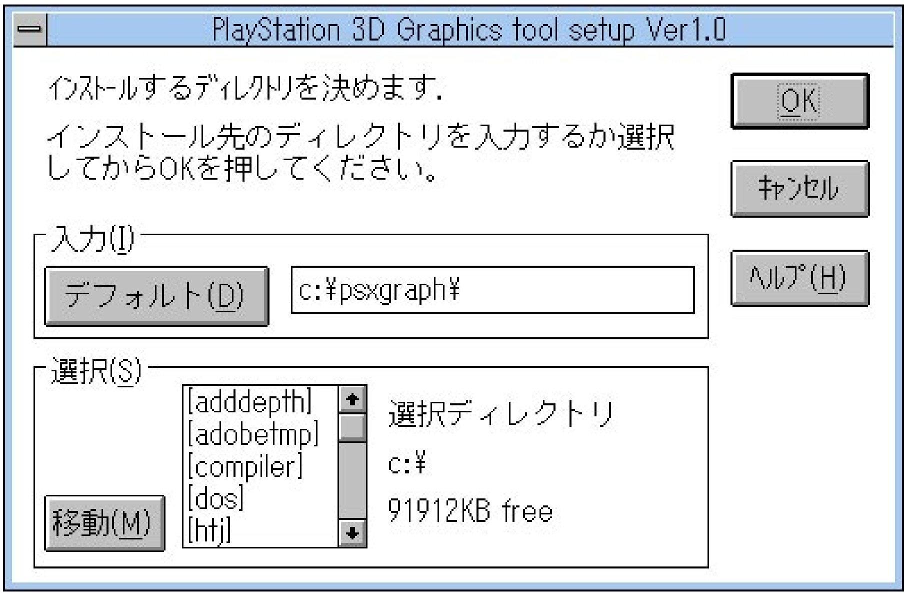


Figure 2-1-3 Selection of Drive and Directory for Installation

Use the default directory as destination of installation. Otherwise, some tools included with 3D Graphics Tool would fail to run normally.

● Exchange disks according to the instructions given by Installer.

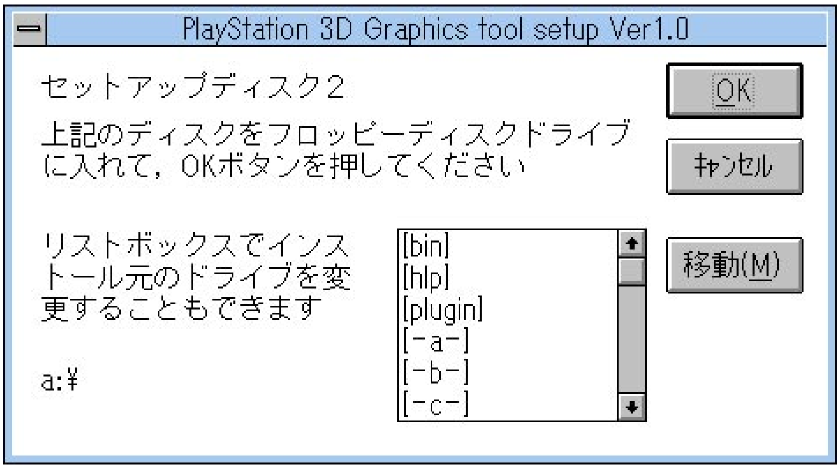


Figure 2-1-4 Exchanging Disks

The above dialogue box may appear even when a correct disk is inserted to the floppy disk drive. If you press the “OK” button, operation for installation will continue.

● When installation is completed, the following dialogue box pops up:

Figure 2-1-5 Completion of Installation

If you press the “OK” button, you are brought back to Windows.

● If you installed “Plug-in for PhotoShop”, copy TIMFMT.8BI and TIMEXP.8BE in C:¥PSXGRAPH¥BIN to PLUGINS subdirectory under the directory where you installed PHOTOSHP.EXE so that PhotoShop can use it.

● Installer does not save groups or icons to Program Manager. If you want to save them, follow the steps described below after completing installation of "Program (EXE.DLL, VBX,HLP, etc.":

1. Select "Saving and Creation of Group" on icon menu of Program Manager.

2. A "Saving of Icon and Creation of Group" dialogue box pops up. Select "Creation of Group", and press the "OK" button.

3. Make the following entries to the "Change saved group data" dialogue box:

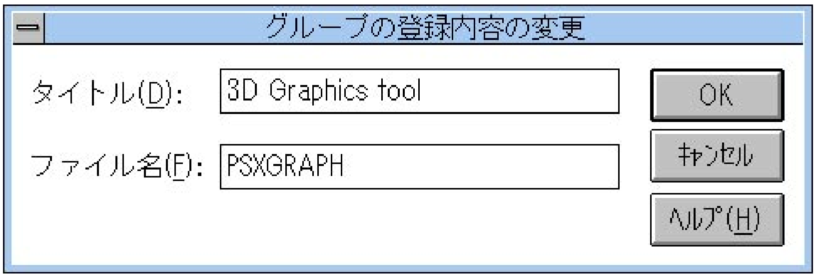


Figure 2-1-6 Determining Group Name

If you press the "OK" button, a new group window is created under the specified title.

4. Open C:¥PSXGRAPH¥BIN using File Manager.

5. Drag the icons for ABOARD.EXE, ANIMATIO.EXE, DXF2RSDW.EXE, MDEDITOR.EXE, AND TIMUTIL.EXE to drop them to the group window prepared in the previous step. Now, necessary programs have been loaded into Program Manager.

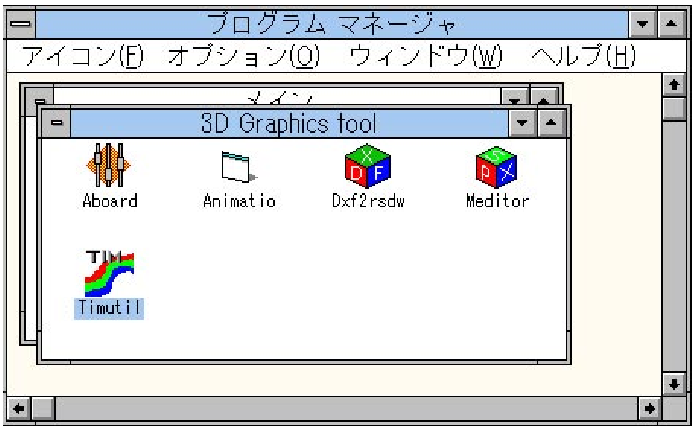


Figure 2-1-7 Saving Programs to Program Manager

2 Download from BBS

If you downloaded 3D Graphics Tool from BBS. A document file describing the installation procedure is attached to it. According to the description, install 3D Graphics Tool into hard disk.

CHAPTER 3

OVERVIEW

This chapter describes the outline of 3D Graphic Tool.

1 Flow of Operation

Editing of 3D graphics data in PlayStation can be classified into the following three stages:

Modeling

You create the shape of a 3D model.

3D Graphics Tool does not include a tool (modeler) for modeling.

“trueSpace for PlayStation” (DTL-S280) available from Caligari Corporation allows you to directly edit 3D model data (RSD format) for PlayStation. If you are going to create model data using any other commercially available modeler, you have to save data in DXF format, and convert it into RSD data. 3D Graphics Tool includes a data converter for that purpose.

Editing material

You set materials, such as colors and textures, on the surface of a model.

3D Graphics Tool includes a tool for material editing.

To use textures as material, you must create image data for textures. You can create image data (TIM) format for PlayStation using Sprite Editor (DTL-S210A) available from Sony Corporation. You can also convert standard image data format (BMP, PICT, RGB) using the attached image data converter. Using the attached PhotoShop plug-in module, you can directly read and write a TIM file in PhotoShop of Windows.

Creating animation

You lay out 3D models properly, and create animation data.

3D Graphics Tool includes tools with which you can create data for a path animation, such as an airplane flying along a certain path, and articular animation, such as a doll of articular construction.

You can also use a tool to create animation data unique to PlayStation called a MIMe animation.

2 Programs Constituting 3D Graphics Tool

3D graphics Tool includes the following programs:

Programs related to image data

Used to process image data.

Program	Function	Environment
BMP2TIM.EXE	Conversion from BMP to TIM format	DOS
PICT2TIM.EXE	Conversion from PICT to TIM format	DOS
RGB2TIM.EXE	Conversion from RGB to TIM format	DOS
TIMEXP.8BE	Plug-in module for TIM data in PhotoShop	Windows
TIMFMT.8BI	Plug-in module for TIM data in PhotoShop	Windows
TIMPOS.EXE	Setting of attribute values for TIM data	DOS
TMUTIL.EXE	Format conversion for various image data	Windows

Table 3-2-1

Programs related to 3D model data

Used to process model data.

Program	Function	Operating environment
DXF2RSD.EXE	Conversion from DXF to RSD format	DOS
DXF2RSDW.EXE	Conversion from DXF to RSD format	Windows
RSDCAT.EXE	Coupling of RSD data	DOS
RSDFORM.EXE	Change of RSD data	DOS
RSD2DXF.EXE	Conversion from RSD to DXF format	DOS
RSDLINK.EXE	Conversion from RSD to TMD format	DOS
TMD2PMD.EXE	Conversion from TMD to PMD (*) format	DOS
TMDSORT.EXE	Optimization of model data	DOS

(*) 3D model data format unique to PlayStation

Table 3-2-2

Programs for editing materials

Used to set materials (surface attributes), such as colors and textures, to a 3D model.

Program	Function	Operating environment
MEDITOR.EXE	Setting of materials	Windows

Table 3-2-3

Programs related to 3D animation

Used to lay out model data, and create animation data.

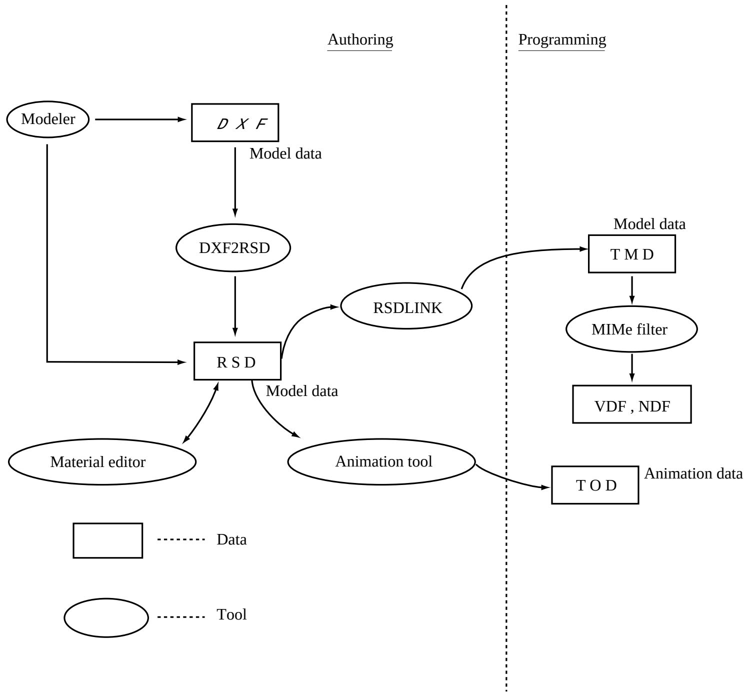
Program	Function	Operating environment
ANIMATIO.EXE	Layout of objects, and editing of animations	Windows
MKTOD.EXE	Creation of TO data	DOS
MIMEFILT.EXE	Creation of MEMe animation data	DOS
MIMESORT.EXE	Optimization of MIMe animation data	DOS

Table 3-2-4

Miscellaneous

Program	Function	Operating environment
ABOARD.EXE	Setting of Artist Board port addresses	Windows

Table 3-2-5



RSD… Defines the shape of a model, and surface attributes.

Used by 3D Graphics Tool.

TMD… Binary RSD format

Built into a program.

TOD… Animation data created by the animation tool.

Built into a program.

TIM… Defines image data.

Used as data for texture mapping.

VDF, NDF… MIMe animation data created by the MIMe filter.

Figure 3-2-1 Flow of Operation of 3D Graphics Tool and Data Format

PART 2

TUTORIAL

CHAPTER 1

OVERVIEW

This chapter describes the outline of Tutorial.

Tutorial explains how to use 3D Graphics Tool, giving many examples and exercises. Use each tool, referring to Tutorial.

Before starting Tutorial, you have to install tools and sample data for Tutorial. For detailed procedure for installation, see CHAPTER 2, gINSTALLATION h in PART I.

You proceed with Tutorial in the following order:

CHAPTER 2 CREATION OF 3D MODEL DATA

CHAPTER 3 CREATION OF TEXTURE DATA

CHAPTER 4 EDITING OF MATERIAL DATA

CHAPTER 5 DISPLAY ON PLAYSTATION

CHAPTER 6 ANIMATION TOOL

CHAPTER 7 MEME ANIMATION TOOL

This is the order which you follow in creation of 3D graphics data.

If you read through Tutorial in this order, you will understand the outline of how to create 3D graphics data. However, each chapter is independent. So, you can refer to any desired chapter from time to time.

CHAPTER 2

CREATION OF 3D MODEL DATA

In this chapter, you perform modeling of some simple shapes using 3D Graphics Tool to understand the entire flow of creation of 3D models. Important points which should bear in mind during modeling are described.

1 Three 3D Model Formats

Let us see file formats related to 3D models.

3D Graphics Tool uses three different formats according to jobs involved in various processes of 3D model creation.

DXF Standard format

Used to create and edit model using a commercially available 3D modeler.

↓

RSD Artist's format

Used for editing with 3D Graphics Tool.

↓

TMD Programmer's format

Used for game programs.

Figure 2-1-1 3D Model Formats

To create characters and buildings used for a 3D game, you perform modeling with a commercially available 3D modeler, convert format using 3D Graphics Tool, add material data, and finally create a file which can be utilized by a program.

DXF format

To create 3D models, such as game characters, you use a graphic tool called the "3D modeler". The standard format called "DXF" exists in 3D models. Most of the commercially available 3D modelers support this format. Thanks to DXF, data can be exchanged with different 3D modelers. 3D Graphics Tool is designed to use DXF so that you may select any desired one from among many 3D modelers. For details of DXF, see "AutoCAD Reference Manual" available from AutoDesk Corporation.

Note that "trueSpace for PlayStation" directly reads RSD format without using DXF.

RSD format

RSD format is mostly used for texture mapping. This format is also used to create an animation. RSD format consists of the following three files:

- File-related data (*.RSD)
- Polygon data (*.PLY)
- Material data (*.MAT)
- Group data (*.GRP)

A model is expressed using these four files. A collection of these files is called “RSD data”, or simply “RSD”. They can be easily edited because they are all ASCII text files. For details of RSD format, see “File Format TMD” in APPENDIX.

TMD format

The TMD format is efficient for PlayStation while lthe RSD format is easy to use for human beings. The TMD is the binary file and thus can be given directly to the library function which is available with “Programmer tool”. It is also a format which is finally printed on CD-ROM. For detail of the TMD format, refer to “APPENDIX: File format; TMD”.

2 Creation and Conversion of Simple 3D Model

Saving of 3D model

First, let us create a simple shape, and convert DXF into RSD. Create a cube using the modeler you have. It may not be a complete cube. Save it as DXF. Saving of DXF differs from one modeler to another. In a certain modeler, the type of the output file may be designated as "DXF" in the dialogue box including a "Save under a different name" command. In another modeler, the data may be saved using the "export" command. For details, see the manual for your modeler.

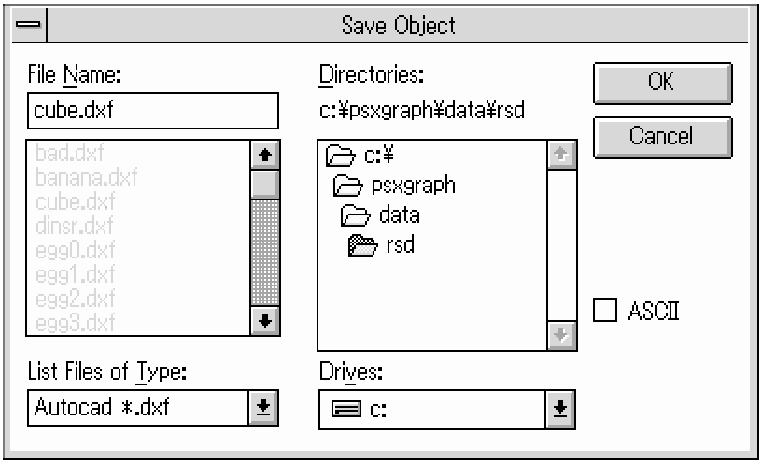


Figure 2-2-1 Save DXF Command

Let us see the directory structure for sample data used in Tutorial. For many samples, subdirectors "RSD" and "TIM" are put as a pair under a common parent directory.

```
¥PSXGRAPH
  └─ TUTORIAL
       ├─ …
       └─ CUBES
            ├─ RSD
            └─ TIM
```

Figure 2-2-2 Example of Sample Data Subdirectors

The name of the parent directory indicates the tile of exercise. The 3D model is saved in the RSD directory, and texture data pasted to the 3D model is saved in the TIM directory. In the following example, exercise is performed under the directory “CUBES”. (In this chapter, however, the TIM directory is empty because no texture is used.)

Now, save the cubic model you created. Save DXF under the directory CUBES¥RSD because it is model data. Name the file as CUBE.DXF. There is a CUBE0.DXT file under CUBES¥RSD. If you do not have a 3D modeler, copy it under the name “CUBE.DXF.”

Conversion of 3D model

Now, let us perform conversion. For conversion, you use DXF2RSD.EXE. Make sure that the MS-DOS prompt is on the screen. Move the current directory to ¥PSXGRAPH¥TUTORIAL¥CUBES¥RSD, and enter the command described below. (The command may be lowercase letters. In Tutorial, filenames and command names are written in uppercase letters.)

```
> DXF2RSD -auto CUBE.DXF
```

Then, four files are created under the directory: CUBE.RSD, CUBE.PLY, CUBE.MAT, and CUBE.GRP.

```
> DIR /W CUBE.*

CUBE.DXF CUBE.RSD CUBE.PLY CUBE.MAT CUBE.GRP
```

Checking 3D model:

Let us load the created RSD data into Material Data, display it. If Artist Board is not yet set, and if Material Editor is not yet installed, perform setting and installation beforehand. (CHAPTER 2, “INSTALLATION”, in PART I, “INTRODUCTION”)

Using File Manager in Windows, start MEDITOR.EXE. Select “Open” on “file” menu, select CUBE.RSD in the CUBES¥RSD directory in the file dialogue, and press the “OK” button.

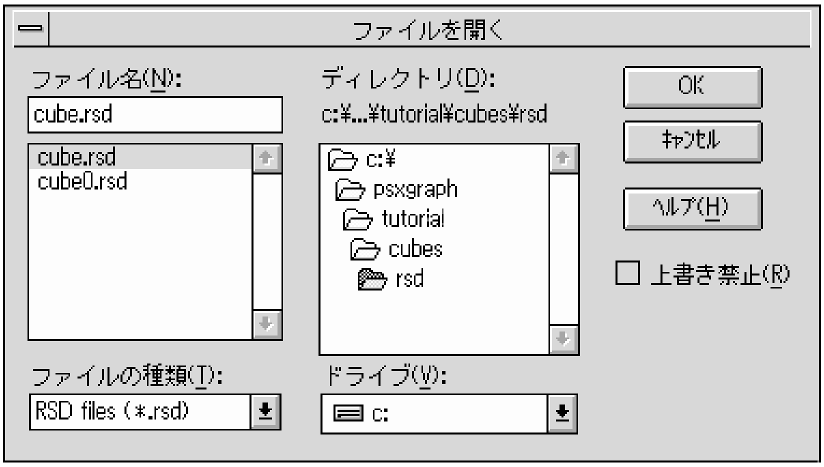


Figure 2-2-3 Loading CUBE.RSD into Material Editor

You will see a white square floating at the center of the video monitor connected to Artist Board. If you hold down the A, W, D, X, S, or C key, it rotates in the three-dimensional direction. You will clearly see that it is a cube to which light is directed. You can zoom in and out it using the “Shift+S” and “Shift+C” key combinations. Use this function as required.

When you have confirmed the cube, select “End” on file menu to quit from Material Editor.

In this chapter, “Check of a 3D model” or “display of a 3D model” means to open a file using Material Editor, rotate the cube to observe the shape, and quit from Material Editor (if necessary). In this chapter, Material Editor is operated only once for the above purpose. Other operations of Material Editor are described in CHAPTER 4 in detail.

In the above operation, an error is displayed on the screen, and DXF2RSD.EXE may be terminated, or the video monitor may fail to display a cube normally. The cause may be an illegal DXF file, or lack of necessary data for conversion by DXF2RSD.EXE. In most cases, such a problem can be solved by proper operation of the modeler, or designation of options in DXF2RSD.EXE. If a cube is not displayed properly although RSD data has been created, read 2.3, “Options in DXF2RSD.EXE”, and make a try. If you cannot create RSD, see 2.1, “Supplementary Description of DXF2RSD.EXE”, in PART III.

3 Options in DXF2RSD.EXE

This section describes some useful and important options. In the following description, [ ] indicates a button or field name applicable to DXF2RSDW.EXE, which is the Windows version of DXF2RSD.

Automatic scaling [automatic size adjustment]

-auto

We used only this option in exercise for a cube in Section 2.2. This option automatically expands and shrinks DXF model data to a size suitable as model data for PlayStation (so that the cube may be put within a cube whose dimension of a side is 1000). At the same time, the model is move parallel to the center of the screen (the zero point of the coordinate system) wherever it may be originally located in the 3D space. Generally, the 3D modeler has its own unit of length. So, a model may be too small or too large to see even if data is simply converted. This function is helpful when you want to check how a model is seen on PlayStation.

If you already know the dimensions and position of the model, you can directly specify the scale factor and amount of parallel translation using the following options:

Scale factor [magnification factor]

-sc factor

Expands or shrinks a model. Specify a scale factor (decimal fraction) in the argument.

Parallel translation [model position]

-t dx dy dz

Moves a model. Specify the amount of movement in the direction of each axis in the argument.

Normal inversion [polygon attribute, normal inversion]

-back

On some modeler, a cube, which was prepared with Material Editor, may look semitransparent when it is displayed, or it may look queer because it is out of perspective when it is rotated. You see both front and back of each polygon because they are not properly defined.

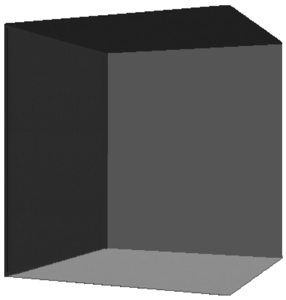


Figure 2-3-1 Cube with Inverted Normal Vectors

DXF does not allow direct description of “front and back of a polygon”. The front is determined according to the order of description of vertexes of a polygon. Depending DXF, this order may be reversed. In such a case, the front and back can be exchanged by specifying this option. The vector which expresses the direction of the front of a polygon is called a “normal vector”.

BACKCUBE.DXF is a cube consisting of polygons whose front and back are inverted. Make sure how the cube looks when the -back option is used and not used.

Coordinate conversion [coordinate system]

-Y-Z -Y+Z +Y-Z +Y+Z -Z-Y -Z+Y +Z-Y +Z+Y

In 3D modeler, you adjust the shape of the entire 3D model while changing the viewpoint to front, rear, right, left, up, and down. If you display a model on PlayStation, the front of the model may be put to the side, or to any other direction.

This is because the coordinate axes are not properly determined by the 3D modeler. Using this option, you can specify the method of conversion of the coordinate system. You specify the coordinate axes at the front and on the top as seen from the front of the coordinate system of the modeler. For instance, "-Y+Z" means that the front is the negative of the Y-axis, and the top is the positive of the Z-axis. Default is "-Y+Z". DXF2RSD.EXE converts them to the coordinate system (-Z-Y: The front is the negative of the Z-axis, and the top is the negative of the Y-axis.) for PlayStation. (Figure 2-3-2)

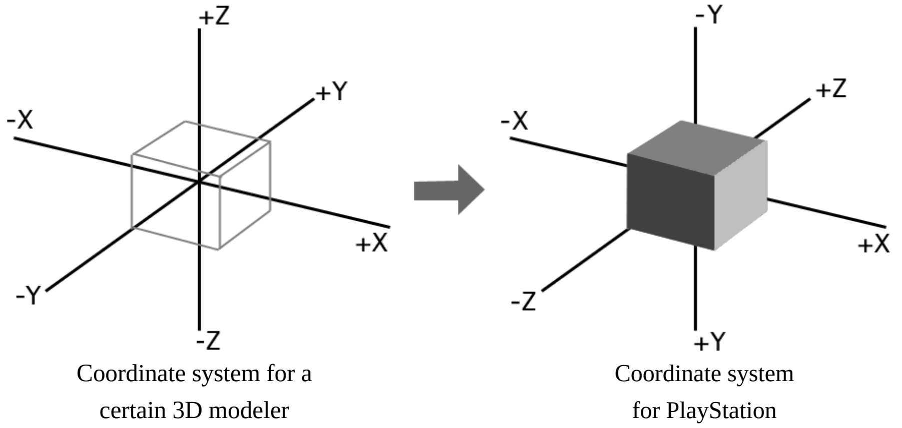


Figure 2-3-2 Conversion from 3D Modeler Coordinate System to Play Station Coordinate System

The front and back of a polygon are inverted each time you exchange Y-Z axis and + and - once using this option. If you specify correctly, you do not have to specify the -back option.

CAUTION: The coordinate system for DXF is described above. It does not always match with the coordinate system on the screen depending on the 3D modeler. Therefore, prepare a 3D model, such as a character, whose top, bottom, right, and left can be easily seen, and find a suitable option so that the front may be displayed when it is loaded from Material Editor.

CAUTION: You do not have to be nervous about coordinate conversion. Even if there is an error in coordinate conversion, you can correct it using the program because the viewpoint can be freely set with the program. However, the front and back must be correctly set.

Quadrangular polygon [polygon shape, quadrangular)

-quad

Generates a quadrangular polygon. If this option is not specified, the model is divided into triangular polygons. You can reduce the number of polygons using this option.

Smooth shading [smooth shading]

-s-g or -s

Used to create model data for smooth shading. Normal vectors are assigned to all the vertexes, and their directions are gradually changed to realize smooth surfaces. If this option is not specified, flat shading ensues, and a normal vector is formed on each polygon, and not on each vertex.

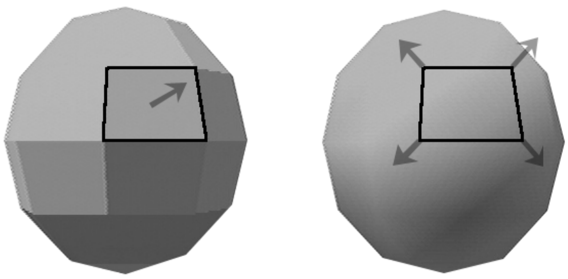


Figure 2-3-3 Flat Shading and Smooth Shading
(Arrows indicate normal vectors related to the quadrangular polygons.)

Smooth shading for a cube is difficult to understand. Make a try using a curved-surface model, such as a ball. (Sample data in SPHERE.DXF is sphere data.)

Data display [data...]

-v

-info

Using these options, you can see the position and size of a DXF model. These options display the types of polygons (triangle, quadrangle, polyline) and their numbers. They are helpful when estimating the approximate quantities of data. The size of a created model is approximately determined depending of the modeler. It will be advisable to know approximate values by executing these options beforehand. (Thus, you can see the scale factor by which a model should be expanded or shrunk.) The -v option performs conversion, and the -info option does not perform conversion.

If you perform conversion using sample data in CUBE0.DXF, and specifying the -V option, the screen display shown in Figure 2-3-4 appears.

```
> DXF2RSD -v CUBE0.DXF
==================================================
 Input DXF file   : CUBE0.DXF
 [DXF]  SIZE      :  204 lines
        VERTEX    :   24
        POLYGON   :   12 (estimate)
                   4-poly   :    6
        RANGE  x  : -250.000 ... +250.000
               y  : -250.000 ... +250.000
               z  : -250.000 ... +250.000
        MATERIAL  :    0

 [RSD]  VERTEX    :    8
        POLYGON   :   12
                    triangles  :   12
        MATERIAL  :    0
        NORMAL    :   12

 Output files     : CUBE0.[rsd, ply, mat, grp]
```

Figure 2-3-4 Sample Output for -v Option

The above sample output indicates that the number of lines (size) of CUBE0.DXF is 204, the number of vertexes is 24, and the number of polygons is equivalent to 12 (converted into the number of rectangles). In reality, the model consists of six quadrangles (4-poly). RSD in the output indicates eight vertexes, and twelve polygons, and that they are all triangles.

For details of the output, see Section 2.1, “DXF2RSD.EXE”, in PART III, “REFERENCE”. For options in DXF2RSD.EXE, also refer to the above description.

Most of the functions of DXF2RSD.EXE can be also used with Windows version DXF2RSDW.EXE. For details, see Section 2.2, “DXF2RSDW.EXE”, in PART III, “REFERENCE”, or online help. Even if free memory space is insufficient in MS-DOS, conversion is sometimes possible if you use DXF2RSDW.EXE.

4 Creation of “Helicopter” and Tips for Modeling

Let us have exercise using a 3D model more worthy of the name “game character”. This section describes important points you should understand when creating a 3D model for PlayStation. Because PlayStation performs all the calculations for drawing real-time, you need special techniques different from a 3D still picture, rendering of which is allowed to take several tens of hours.

Let us prepare a helicopter as shown in the figure below, and convert it into RSD. In the subsequent chapters, RSD data will be used to apply colors or patterns, and make the window of the cockpit semitransparent.

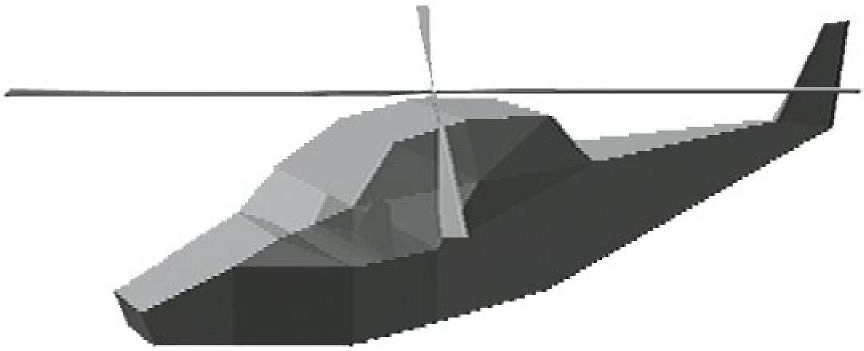


Figure 2-4-1 View of Finished Helicopter

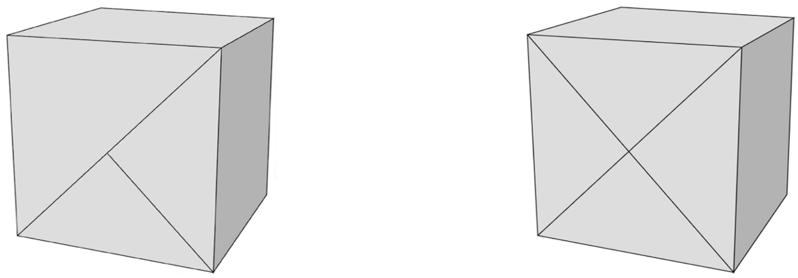
This helicopter consists of two objects: body and rotor. They can move independently, and hierarchical relations between them can be defined. For these purposes, however, each object must be created as different RSD data. (In other words, the parts which move in concert with each other should be defined as one object and coordinated into the same RSD even if they are apart from each other.

Details will be discussed in CHAPTER 6, “ANIMATION TOOL”. Here, you should understand that an object is a part moving independently, and that each object corresponds to its own RSD (and DXF, consequently).

Let us prepare the body. You perform exercise under the HELIMAKE directory. Key points in modeling are described below. Prepare the body according to the following description.

Principle of common vertexes

A side, excepting its vertex, of a polygon should not contact any other vertexes. In such a case, the polygon must be divided so that such vertex may be a vertex of all the adjacent polygons. This is called the principle of a common vertex.



Cube having a non-common vertex　　　　Cube consisting of common vertexes only

Figure 2-4-2 Non-Common and Common Vertexes

If a vertex of another polygon is located on a certain polygon as seen in the right-hand view in Figure 2-4-2, a gap will be created among polygons as shown in Figure 2-4-3. No problem will arise in case of a 3D still picture. However, calculation of coordinates of a point located on the straight line connecting the two vertexes involves errors because coordinate values for a 3D model handled by PlayStation are calculated real-time. As a result, a gap is created among the sides adjacent to each other. On the contrary, if a side of one polygon is in contact with, and coincides with, a side of another polygon (that is, both of these sides have the two same vertexes at both ends) as shown in the right-hand view in Figure 2-4-2, no gap is created because no error is caused for any sides which are in contact.

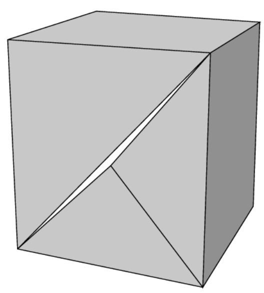


Figure 2-4-3 Gap Caused by Non-Common Vertexes

Also, avoid coupling two 3D models prepared independently from each other by mating the surface of one polygon to that of another as shown in the figure below. No problem will arise if the vertex coordinate values are just the same. Otherwise, a gap will be created. The two coupled surfaces are useless polygons which are never seen.

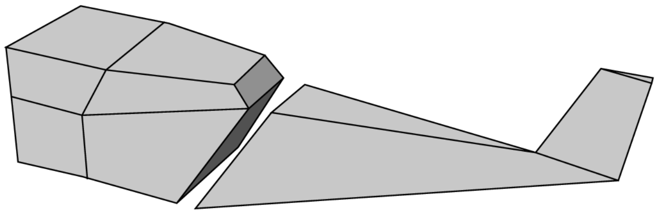


Figure 2-4-4 Improper Sample (Pasting of Polygons

Using the modeler, you should extrude an existing surface or side to create a new surface instead of inserting new vertexes to a middle point of a side, or to any other position (even if you can do so). This function is one of the fundamental functions of the 3D modeler, which is called “Extrusion”.

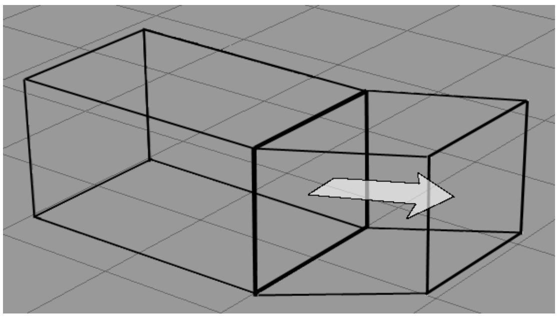


Figure 2-4-5 Extrusion of Surface

Otherwise, a fundamental shape, such as a rectangular parallelopiped, may be divided into small polygons, and then the shape may be modified by moving the vertexes.

In this tutorial, the extrusion function (sweep and slice) of the 3D modeler called “”trueSpace for PlayStation” is repeatedly used to prepare the helicopter.

HELI1.DXF through HELI5.DXF in the HELIMAKE\RSD directory include such data.

Number of Polygons

The surfaces of the helicopter included in sample data are relatively rough. The 3D modeler can be used to change its shape into a one having many curves using the 3D modeler. To express a curve using polygons, many small polygons are needed. If too many polygons are used, a great burden will be imposed on drawing, and missing frames are likely to result. The person who prepares a model must know how many polygons can be used.

When preparing a certain scene in a game, determine the number of polygons used for each object considering the balance of the entire screen. The main character will need many polygons, and buildings in the background should consist of fewer polygons. The performance (maximum 360,000 polygons without texture per second) of PlayStation allows use of about 12,000 polygons for display of 30 frames per second, and about 6,000 polygons for display of 60 frames per second. Less polygons must be if textures are employed, or if smooth shading is performed. The number of polygons is also greatly affected by the size of polygons on the screen, and program composition. It is advisable to prepare a model so that it may be consist of 2,000 to 3,000 polygons per frame. Based on this, find the number of polygons used with each object. A game having very beautiful screen generally uses such a number of polygons. To get beautiful screen display, it is more effective to minutely prepare textures than increase the number of polygons. Rough polygons can look smooth if smooth shading is performed. (For instance, display HELI5.DXF using DXF2RSD.EXE with the -g option. You will see a fairly round shaped object.

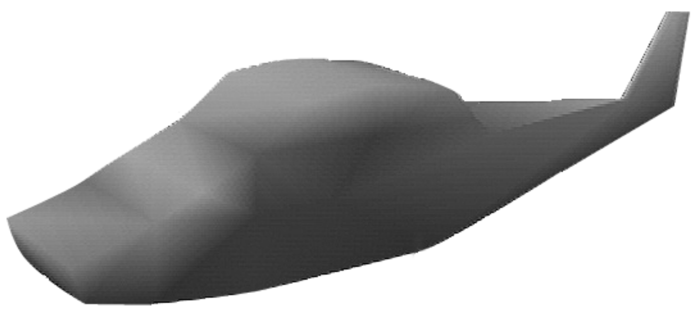


Figure 2-4-6 Helicopter Subjected to Smooth Shading

Front and Back of Polygons

The canopy of the cockpit will be made semi-transparent in CHAPTER 4. The procedure is described in that chapter. Let us display semi-transparent sample data HELI5T.RSD using Material Editor.

If you turn the helicopter, you will see the canopy semi-transparent. If you seen it from the front, the tail of the helicopter. It is seen through the wall at the rear of the cockpit. You are seeing the “back” of the wall behind. When you see from the side, you cannot see the inner wall at the side on the end opposite to the canopy at the front. In the 3D world of PlayStation, the back of a polygon does not reflect light at all. So, you see nothing there. As a result, you will see a hole in the body.

To prevent this, you must insert a front-faced polygon into the inner wall of the cockpit. As an example, let us create the cockpit.

You should learn by heart the equation “Front of polygon = Direction of normals of polygon = Seen”. If there is a polygon which is observed in the wire frame but, which are not displayed, you should check to see if there is a reversed polygon.

Creating the Cockpit

Let us put a cockpit to the body. Unless the wire frame is used, you cannot see polygons at the inner part (inside) of the 3D model, such as the cockpit. However, you can see it through the canopy when the canopy is made semitransparent using Material Editor in CHAPTER 4.

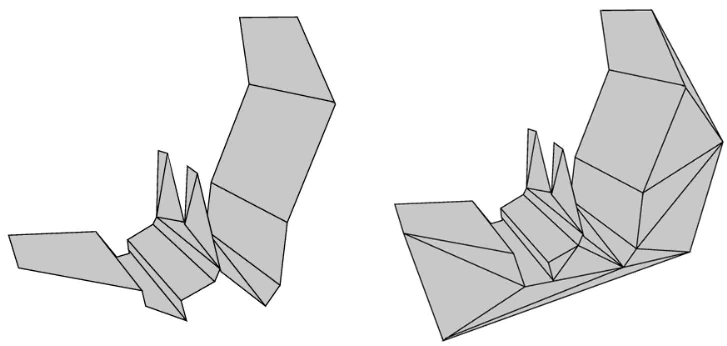


Figure 2-4-7

It will be easier to prepare the cockpit separately from the body, and paste it later. You should use care when preparing the shape of the cockpit. The left-hand view above shows the cockpit consisting of a folded sheet, which has no volume. Some modelers (solid modelers) do not allow you to prepare such a shape. In that case, the shape of one side of a rectangular parallelopiped as shown in the right-hand view may be changed. HELI6.DXF is the model which use the cockpit shown on the right-hand view. In HELI6T.RSD, the canopy is made semi-transparent.

The watt at the side of the cockpit can also be prepared in the similar manner. You can easily crate it using “Copy button” and “front-back reverse button” in Material Editor. For details, see CHAPTER 3, “MATERIAL EDITOR”, in PART III, “REFERENCE”.

You cannot see the cockpit until you make the canopy semi-transparent in CHAPTER 4. HELI9.DXF includes a model to which the cockpit is added. In HELI7.RSD, the canopy is made semi-transparent.

Z Sort Problem

Using Material Editor, check HELI6T.RSD, which is described in the previous section. If you rotate the object, you will see polygons on the sides and bottom of the body flickering. This is the disturbance of drawing, which is called “Z Sort”, ascribable to the method of polygon drawing which PlayStation employs. Z sort is a method used to determine the order of drawing of polygons in the order of depth. The position in the direction of the depth of the center of gravity of each polygon is calculated, and the polygon concerned is drawn if it judged to be located at the front of the already drawn polygons. Z Sort is suitable for real-time drawing because it is more efficient than other sort methods.

Disturbance of drawing due to Z sort occurs when two polygons are pierced into each other, or they are closely located. This is because the results of calculation of the position in the direction of the depth is affected by the viewpoint and direction so that the polygons behind overwrite the polygons at the front. This is called “Z sort problem”, which a person who creates a 3D model must bear in mind.

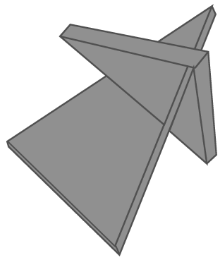


Figure 2-4-8 Z Sort Problem

Measures to solve the Z sort problem are described below. Some measures may be employed during modeling, and others may be adopted for programming. Take suitable measures after consulting with the person responsible for the entire program.

- Change the way in which model data are created. Give minute consideration to the shape and layout of polygons by putting apart two polygons from each other, etc. Avoid using thin and long polygons because they are likely to cause the Z sort problem. Sometimes, you can get around the Z sort problem by using quadrangular polygons.

- Increase the size of the polygon causing Z sort. If you simply use smaller polygons, the number of polygons will increase. So, you should pay attention to trade-off.

- Increase the size of the ordering table (OT) used for the program. Thus, you can improve the precision of Z sort. You can set the size of OT using the screen display of Material Editor. However, the load will increase as the size of OT increases.

- Explicitly describe the order of drawing in the program. As when an automobile runs on a road, there will arise no Z sort problem concerning the road and tires if you always draw the road first. This method is very effective when the order of drawing is fixed as in the above example.

HILI8.DXF includes an improved helicopter sample. In HILI8.RSD, the canopy of the cockpit is made semi-transparent. In this sample, the shape and size of polygons near the cockpit are changed. Make sure how they are seen on Material Editor. If you encounter a Z sort problem in your model, and if you want to solve it, make a try to improve the model.

Near-Clip Problem

As the viewpoint closely approaches a polygon, it may become too large to display on the screen. This occurs when you enlarges a polygon by continuously holding down "Shift + X" on Material Editor. This is called "near-clip". As in a racing game in which the viewpoint moves along the road, the portion near the front may be cut off, spoiling the scene.

To avoid this, it is effective to use the automatic split function of the library for the program. However, this function causes a great load during execution of the program. If the portion where near-clip occurs is know, the polygons at that part may be divided further to alleviate the inconveniences.

Hierarchical Structure

If you put the rotor to the body you created, the helicopter will be completed. Save the data in DXF, and convert it into RSD by running DXF2RSD.EXE. Because the body and rotor are separate objects, save their data to different DXF. For safety, let us save the entire data. At this time, data may be in DXF or in any other format. Data may be in any format as long as it can be loaded whenever necessary to do the same job again. (It is a good practice to save data under different names during modelling. It can happen very often that you find a defect during editing, and it is impossible to perform UNDO when you notice it.)

To save objects separately, you may follow these steps:

1) With all the objects displayed on the screen, select all the objects, excepting the one which you want to save.

2) Delete the selected objects.

3) Give a new name to the remaining object, and save it as DXF.

4) Restore all the objects by performing UNDO or loading the file. Return to 1), and save the next object.

Separately save the body and rotor as “HELI.DXF”, and “ROTOR.DXF”.

Some 3D modeler has the function to save a selected object. Check this by referring to the manual for your 3D modeler.

When all the objects have been saved, convert them into RSD by executing DXF2RSD.EXE.

```
>DXF2RSD HELI ROTOR
```

There are sample data HELI8.DXF and ROTOR0.DXF. Copy them if necessary.

When all the necessary RSD data have been prepared, add surface attributes to them. In CHAPTER 4, textures will be pasted, and windowpane will be installed on the canopy to make the model real.

CHAPTER 3

CREATION OF TEXTURE DATA

This chapter describes how to create texture data using the TIM utility.

You can edit texture data used in PlayStation in file format called TIM. A file in this format can be handled by Sprite Editor (SPRITE.EXE). TIM data can be created by converting data prepared by various event tools available from Windows and Macintosh.

For these purpose, BMP2TIM.EXE, PICT2TIM.EXE, RGB2TIM.EXE, and TIMUTIL.EXE are used as converters. A plug-in module which allows reading and writing of TIM files in Adobe PhotoShop is also available. (See CHAPTER 1, “4 TIMEXP.8BE, and 5 TIMFMT.8BI, in PART III, “REFERENCE”.)

How to convert using the conversion utility “TIM Utility” (TIMUTIL.EXE) is explained below.

TIM Utility allows use of BMP, which is generally used in Windows, PICT used in Macintosh, general-purpose format RBG, and TIM for PlayStation. It converts one format to another, and allows setting of various parameters.

1 Example of Conversion into TIM

Conversion of sample data (¥PSXGRAPH¥TUTORIAL¥TIMUTIL¥TIM¥A.BMP)

This section describes the procedure for fundamental format conversion using sample data.

Now, start, TIM Utility (TIMUTIL.EXE) using File Manager. Select “Open...” on the file menu of TIM Utility. The next dialogue box pops up.

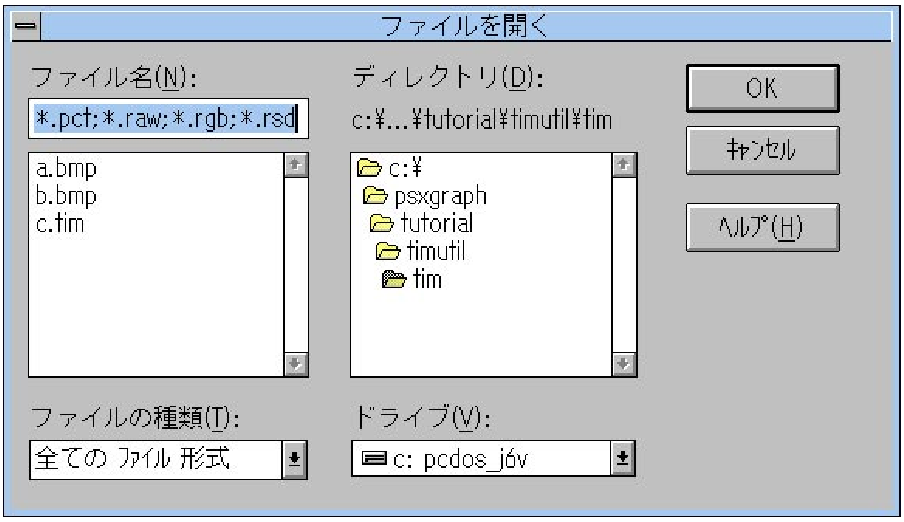


Figure 3-1-1 Selecting a File before Conversion

If necessary, move the directory to ¥PSXGRAPH¥TUTORIAL¥TIMUTIL¥TIM, select “a.bmp”, and press the “OK” button. Then, the following window opens:

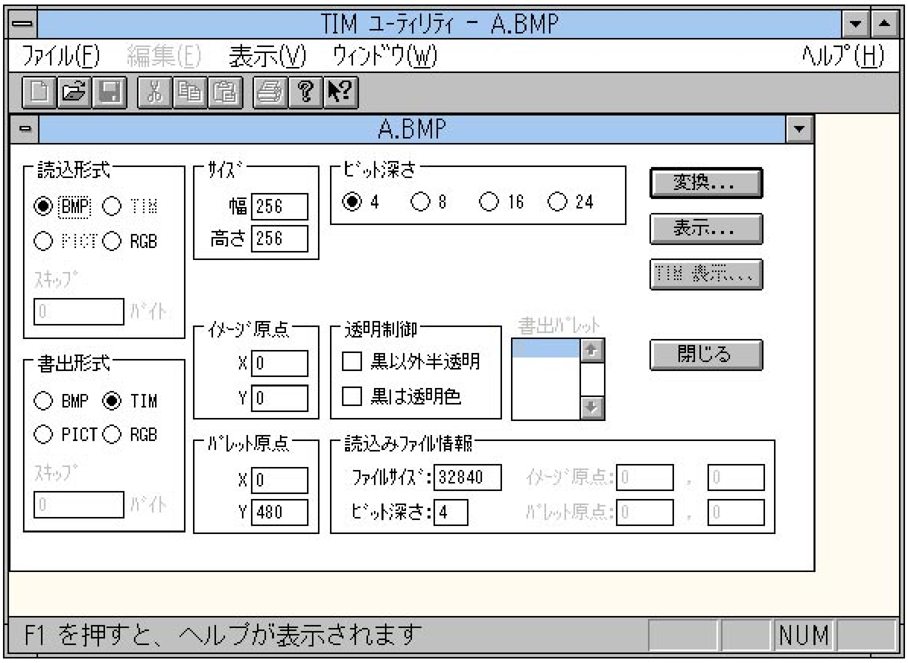


Figure 3-1-2 TIM Utility Opening A.BMP

Let us check the contents of the A.BMP file. Press the "Display..." button. The following window appears.

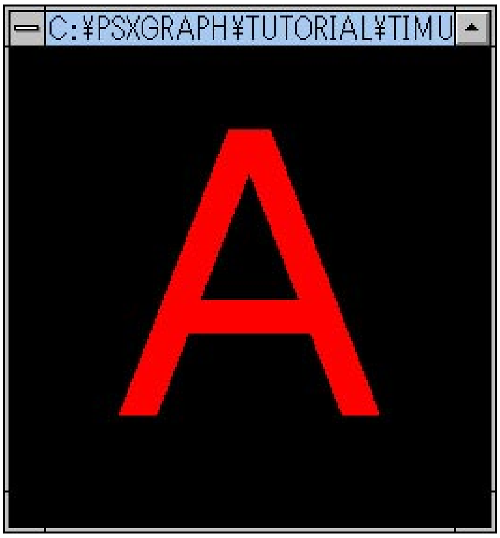


Figure 3-1-3 Displaying A.BMP

This window displays the image contained in the A.BMP file. (You can close this window by pressing the "Esc" key, or selecting "Close" on the menu at the left corner of the window.)

Now, let us convert this image file in BMP format into TIM format. Make sure that "Writing format" is "TIM". (If the file is in any other format, select "TIM" using the left button of the mouse. Press the "Convert..." button, and the following dialogue box pops up.

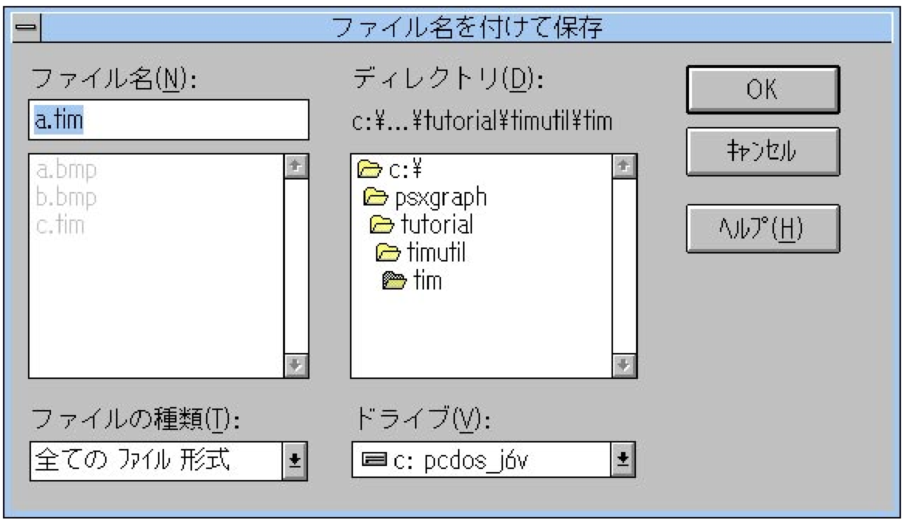


Figure 3-1-4 Specifying a Filename after Conversion

Press the "OK" button, and conversion from BMP to TIM format starts. During conversion, you see the following dialogue box on the screen:

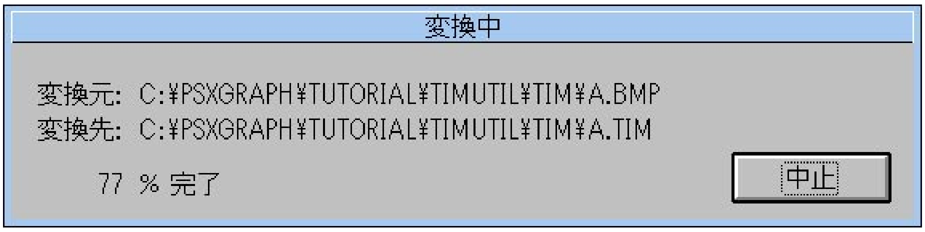


Figure 3-1-5 Dialogue during Conversion

When "100% completed" is displayed, conversion is over.

Now, a TIM format image file ¥PSXGRAPH¥TUTORIAL¥TIMUTIL¥TIM ¥A.TIM has been created.

Confirming the results of conversion

Let us open the ¥PSXGRAPH¥TUTORIAL¥TIMUTIL¥TIM¥A.TIM file. You may open ¥PSXGRAPH¥TUTORIAL¥TIMUTIL¥TIM¥A.TIM by selecting "Open..." on the file manu as you did to open A.BMP. However, let us use the drag and drop function of File Manger. Using File Manger, display the files under the ¥PSXGRAPH ¥TUTORIAL¥TIMUTIL¥TIM directory.

Figure 3-1-6 File Manger for Windows3.1

Select “a.tim” on the window using the left button of the mouse. While holding down the left mouse button, drag it to the window for TIM Utility. When the mouse cursor is put into the window for TIM Utility, release the button. Now, A.TIM is opened.

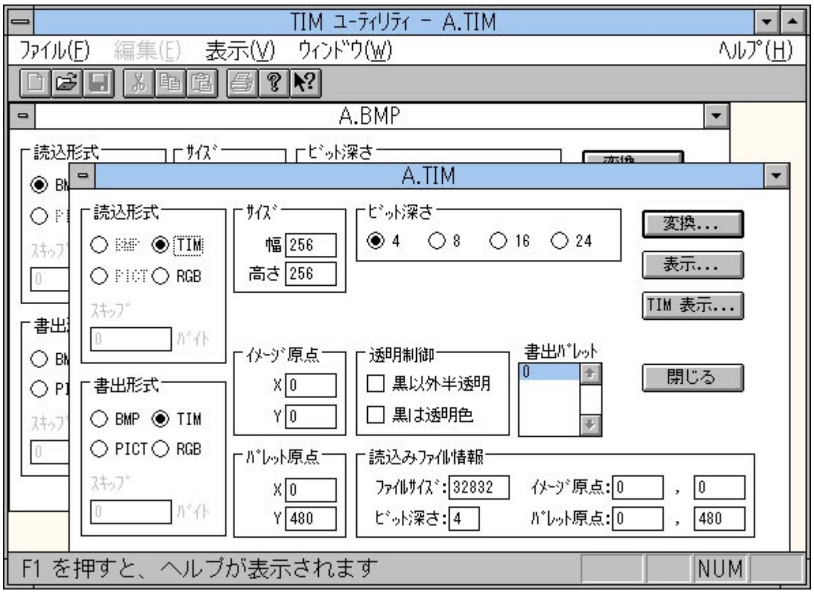


Figure 3-1-7 TIM Utility Opening A.TIM

Let us check the results of conversion. You can display the results using the “Display...” function as you did for A.BMP”. However, images displayed by this function may a little differ from those actually displayed by PlayStation in tints, etc. So, let us use the “TIM display...” function which can only be used when a file of TIM format is opened. If you press this button, you can display a TIM image on the video monitor connected to Artist Board. Press the “TIM display...” button. The following dialogue box pops up, and the image is displayed.

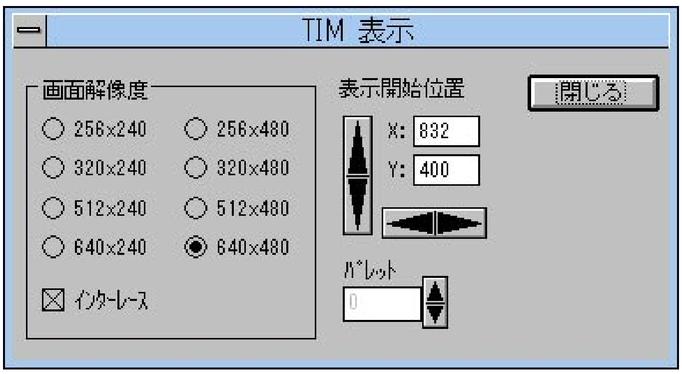


Figure 3-1-8 Display on Artist Board

Using this dialogue box, you can specify the image resolution of Artist Board, and display area start position on VRAM. You can scroll the image up and down, and to right and left by changing the display start position.

Now, you understand how to convert A.BMP into A.TIM, and how to check the results. The following section explains the other functions of TIM Utility.

2 Transparency Control

You can specify an attribute called “transparency control bit” for a TIM format image file for PlayStation. Utilizing this function, you can dig a hole in a polygon, and give an effect of semi-transparent glass using textures. By making the best use of it, you can express a complicated shape using a small number of polygons.

Example: Broken canopy

Using TIM Utility, open ¥PSXGRAPH¥TUTORIAL¥TIMUTIL¥TIM\B.TIM. Set “X” in “Image zero point” to 640, and check the two check boxes for “transparency control”.

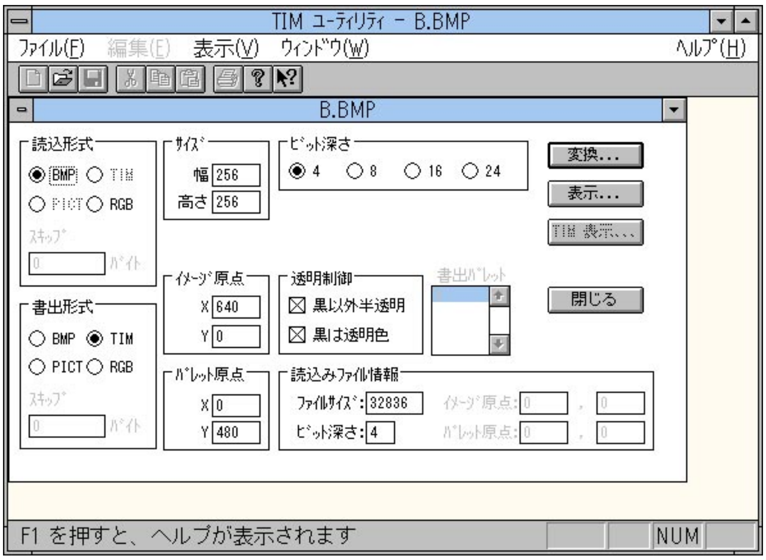


Figure 3-2-1 TIM Utility Opening B.BMP

Make sure that all the items are set as shown in the above figure. Press the “Convert...” button. Be sure that the filename is B.TIM.

Now, start Material Editor (MEDITOR.EXE) using File Manager. (For details of Material Editor, see CHAPTER 4.) Select “Open” on the file menu of Material Editor to open ¥PSXGRAPH¥TUTORIAL¥TIMUTIL¥RSD¥EX1.RSD. The “texture layout” dialogue box pops up.

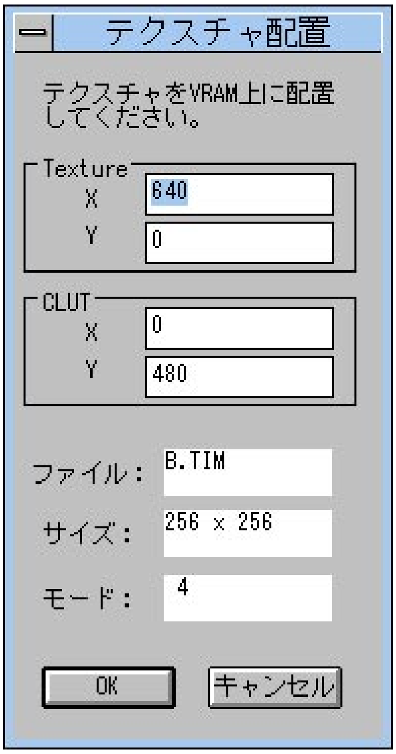


Figure 3-2-2 Texture Layout Dialogue (Material Editor)

Make sure that “X” in “Texture” is set to 640, and press the “OK” button. The following image appears on the video monitor connected to Artist Board:

Figure 3-2-3 Screen display 1 of RX1.RSD with Material Editor

You can rotate the displayed model to up and down, and to right and left using the "W", "X", "A", and "D" keys on the keyboard. You will see a red ball through semi-transparent blue, and through a tear at the center.

Such effect is obtained because you checked "Semi-transparent colors excepting black", and "Transparent black" in "Transparency control" on conversion from B.BMP to B.TIM. The texture image (B.BMP) for this example is as follows:

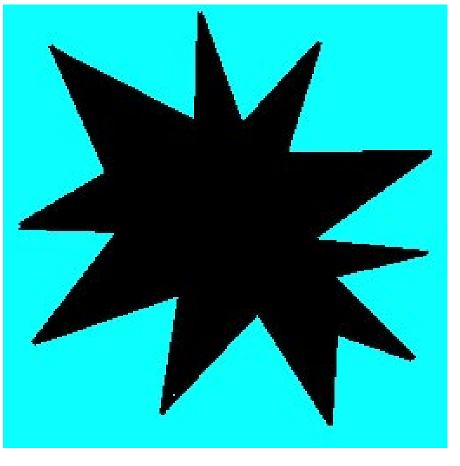


Figure 3-2-4 B.BMP

The tear at the center is painted in black (R, G, and B: All 0), and all the other portions are painted in light blue. So, the rent is made "transparent", and the surrounding portions are made "semi-transparent" when B.BMP is converted into TIM format. If B.TIM is generated without checking “Semi-transparent colors excepting black”, the following screen display appears when ¥PSXGRAPH¥TUTORIAL ¥TIMUTIL¥RSD¥EX1.RSD is displayed using Material Editor:

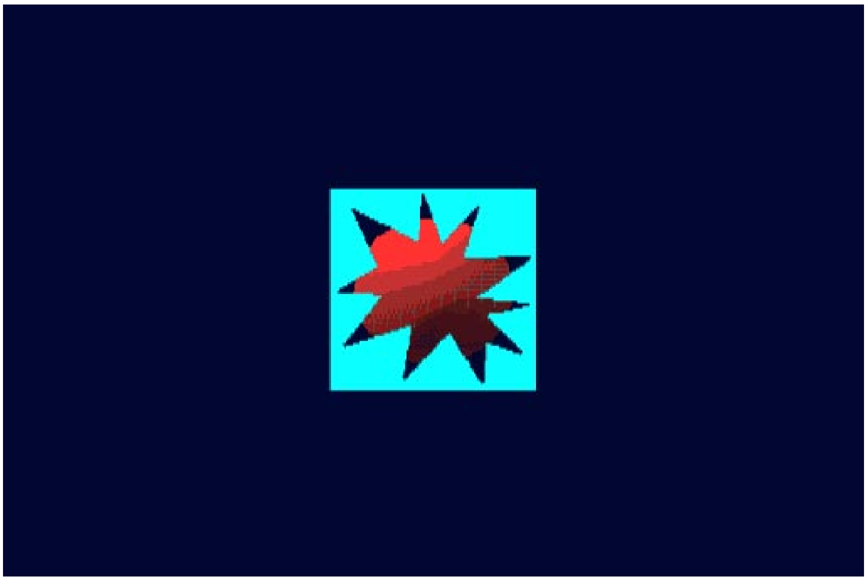


Figure 3-2-5 Screen Display 2 of EX1.RSD with Material Editor

The light blue portion, which looked transparent in the example in which “Transparency control” was checked, now looks opaque.

Restrictions

TIM Utility of the current version performs very rough transparency control for black and other colors. If you want to set transparency control bits for each color on the palette concerning TIM consisting of less than 8 bits, you may use “Sprite Editor” (SPRITE.EXE).

3 Layout of Textures

Textures and palette (CLUT) areas used by PlayStation must be so laid out that they may not be overlaid on other textures, palette, drawing/display areas. To aid such a job, TIM Utility is equipped with a function which graphically lays out textures and palettes.

Re-layout of sample data

Open A.TIM and B.TIM under the ¥PSXGRAPH¥TUTORIAL¥TIMUTIL¥TIM directory, which were created above, and open the already created C.TIM.

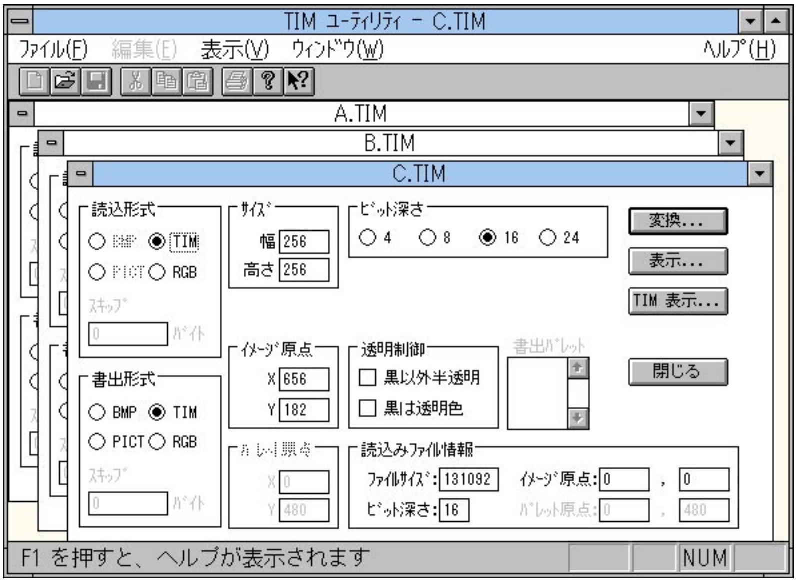


Figure 3-3-1 TIM Utility Opening Three TIM Files

Now, select “TIM layout...” The following dialogue box pops up.

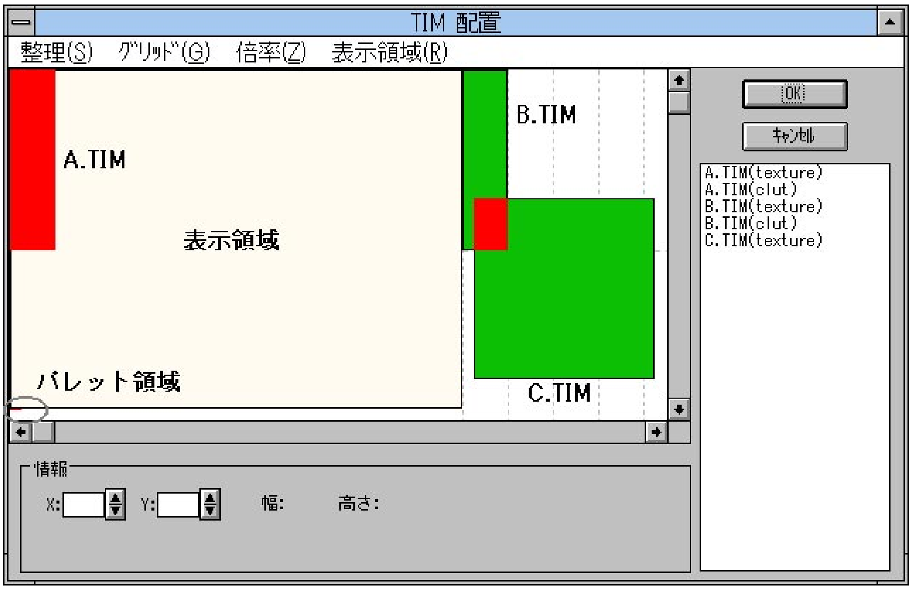


Figure 3-3-2 TIM Layout Dialogue 1

The above figure shows the areas on VRAM occupied by images and palettes in A.TIM, B.TIM, and C.TIM. The broken lines indicate the “texture page boundary”, which will be discussed later in “Cautions in texture layout”. You will see some portions displayed in red (printed in the darkest gray on the paper). They indicate that a problem will arise when textures are actually displayed by PlayStation. A.TIM is located on the drawing/display area. C.TIM is overlaid on the area of B.TIM. The palette areas for A.TIM and B.TIM are also overlaid. They must be moved to such positions that they may not be overlaid on each other. We must move each texture area to a position along the “texture page boundary”.

Let us move A.TIM first. Select the area for A.TIM using the left button of the mouse. While holding down the mouse button, drag it to the right. When it is moved to the right side of B.TIM, release the mouse button.

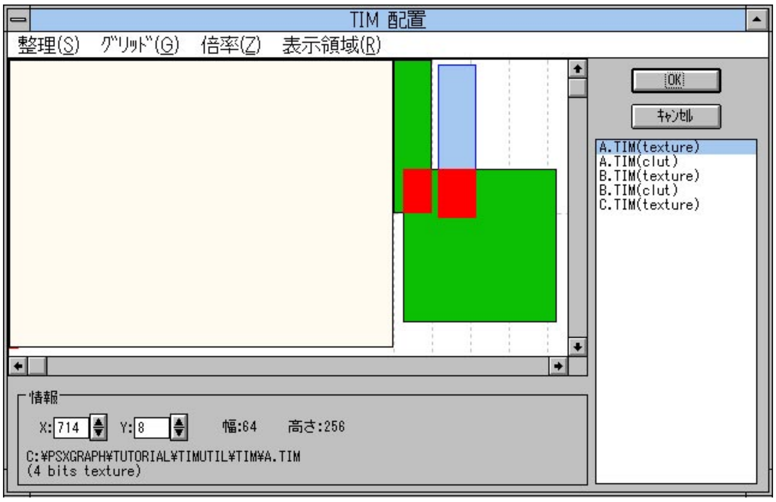


Figure 3-3-3 TIM Layout Dialogue 2

You will see screen display shown in the figure above. It is difficult to precisely put A.TIM onto the texture boundary using the mouse. There are two methods to get around the difficulty. One method is to set numerical values in “X” and “Y” for the “data” area to determine the position. The other method is to utilize the “grid” function. We will use the second method because it is more convenient.

First, select “Texture page” and “XY” on the grid menu. Using the mouse, drag A.TIM so that the left top corner of A.TIM may be put near the right top corner of B.TIM. Release the mouse button, and A.TIM moves to a position along the texture page boundary. In this status, the screen display looks as follows:

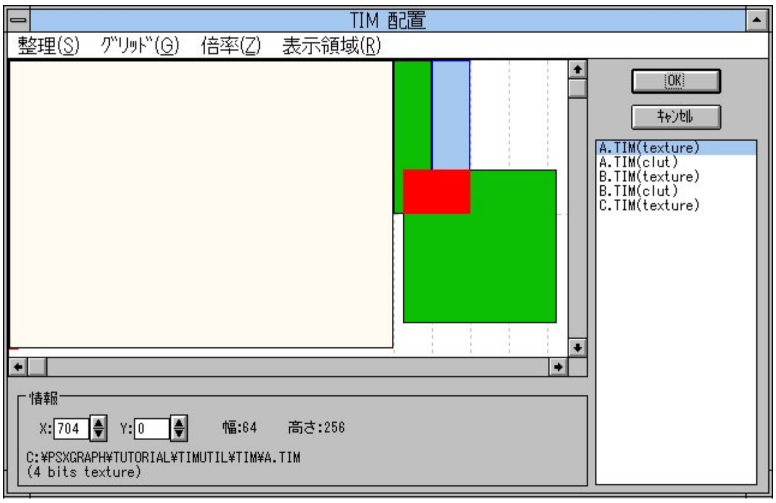


Figure 3-3-4 TIM Layout Dialogue 3

Now, A.TIM and B.TIM are moved to desirable positions. However, C.TIM is overlaid on A.TIM and B.TIM. Looking at the screen, you will see the lower half of VRAM empty. Move C.TIM to the left bottom as you did to move B.TIM. At this time, move the left corner of C.TIM near the left bottom corner of B.TIM. The result will be as follows:

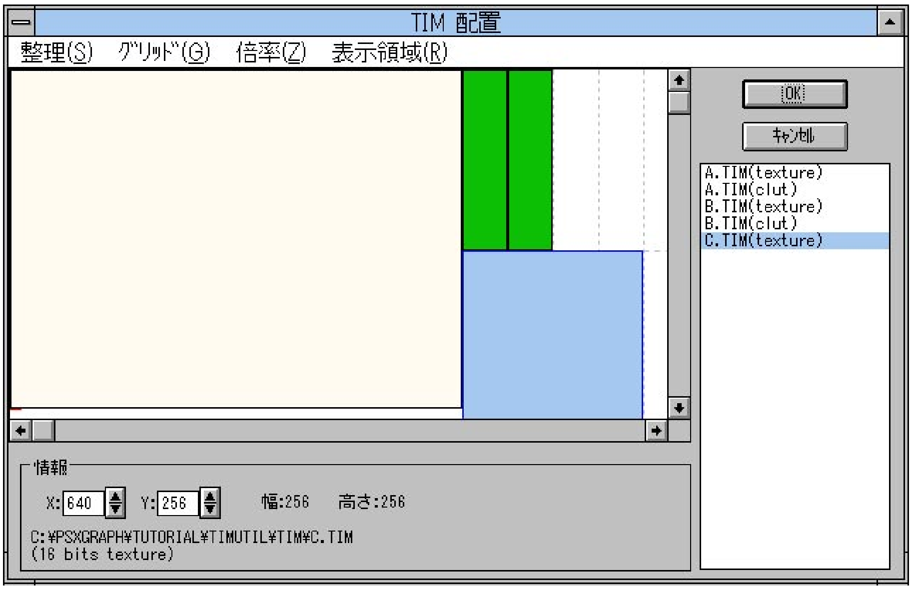


Figure 3-3-5 TIM Layout Dialogue 4

Now, red portions have almost disappeared. If you closely look, there is a red portion in the palette area. Palettes in A.TIM and B.TIM seem to be overlaid on each other. (C.TIM has no palette because it is a 16-bit texture.) To relocate the palettes, you may use the same method which you used to move the image area of the texture, but there is a simple method.

Select "Palette" on the arrangement menu. This function puts all the palettes vertically as seen from the lower end of the display area. Make sure that the palette area which was displayed in red has changed into green.

Now, the position data on the defective texture has been corrected. Press the "OK" button, to return to main window of TIM Utility.

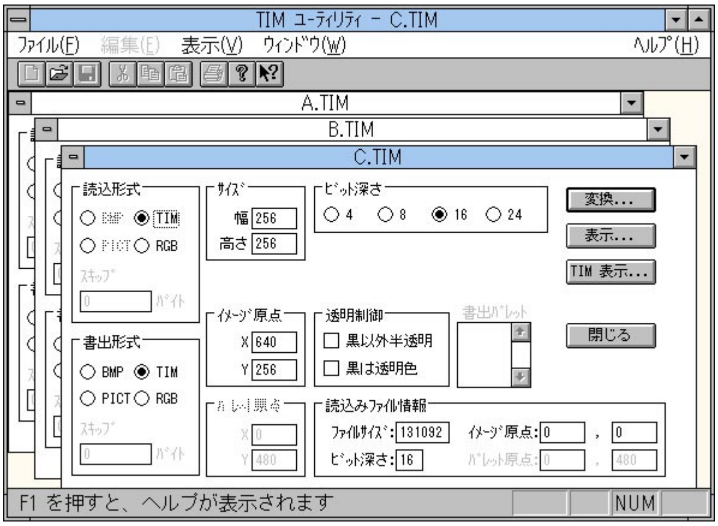


Figure 3-3-6 TIM Utility Completing TIM Layout

Make sure that the values of the “image zero point” and “palette zero point” have been changed by “TIM layout”. In this stage, however, the changes to the “image zero point” and “palette zero point” are not reflected in the TIM file. To do so, you may press the “Convert..” button to save each file. There is a better method. Select “Save all files...” on file menu. The following dialogue box pops up.

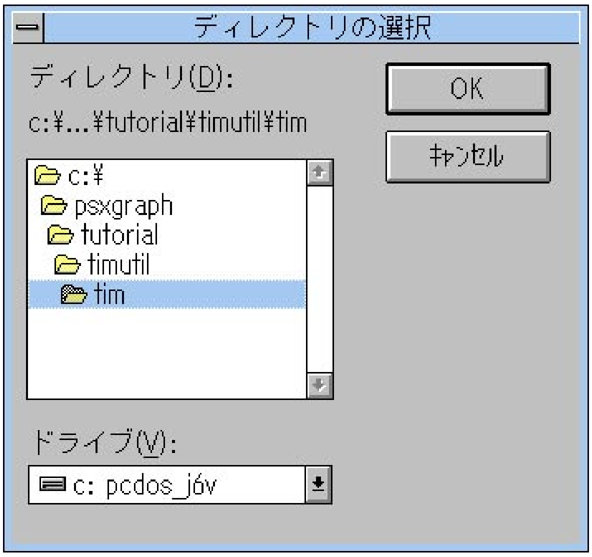


Figure 3-3-7 Dialogue for Directory Selection

Make sure that the displayed directory is the same as that shown in the above figure. Press the “OK” button.

Data on the “image zero point” and “palette zero point” is written to A.TIM, B.TIM, and C.TIM. The following dialogue box appears, and writing of the “image zero point” and “palette zero point” is completed.

Figure 3-3-8 Zero Point Writing End Dialogue

Cautions in texture layout

In VRAM of PlayStation, the position where texture data can be put is restricted by the boundaries set at intervals of 64 words in the X direction, and at intervals of 256 lines in the Y direction. These boundaries are called “texture page boundaries”, and the area demarcated by these boundaries is called a “texture page”.

When actually pasting textures to polygons, UV values (0 through 255), which are offset values, or the numbers of pixels from the texture page and texture page boundaries described above, determine texture data.

If texture data is placed just along the texture page boundary, a texture consisting of up to 256 x 256 pixels can be displayed. If the texture data is put aside from the texture page boundary, the size of the texture will be smaller.

In the transverse direction, data must be put within one texture page if it is a 4-bit texture. In case of an 8-bit texture, it must be within two texture pages. A 16-bit texture must be put within four texture pages. In the vertical direction, texture data must be so arranged that it may be always put within one texture page.

● Improper example

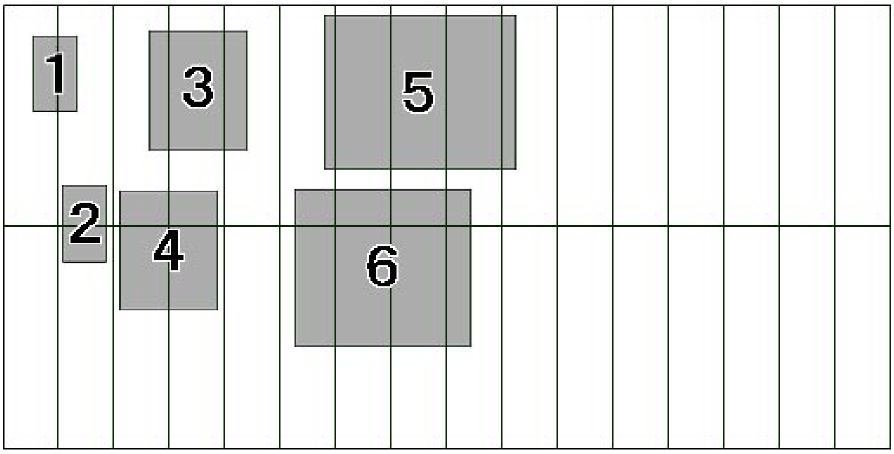


Figure 3-3-9 Bad Examples

1 A 4-bit texture. Vertically, it is put within one texture page, but it transversely crosses the texture page boundary.

2. A 4-bit texture. Transversely, it is put within one texture page, but it vertically crosses the texture page boundary.

3. An 8-bit texture. It is transversely put within one texture page, but it vertically crosses three texture pages.

4. An 8-bit texture. It is vertically put within two texture page, but it transversely crosses the texture page boundary.

5. A 16-bit texture. It is transversely put within one texture page, but it vertically crosses four texture pages.

6. A 16-bit texture. It is transversely put within four texture pages, but it vertically crosses the texture page boundary.

4 Supplement

- You cannot read compressed BMP, JPEG, compressed PICT, 32-bit PICT, and PICT, which does not include any bit map.

- You cannot write compressed BMP, JPEG, and compressed PICT format. The depth of bits in the write file is restricted to 4, 8, 16, and 24 bits.

- If conversion is performed to reduce the bit depth (for instance, from 16-bit data to 8-bit data), color data are approximated. In this case, the precision of colors will be impaired because color maps to which R, G, and B are evenly assigned are used for approximation instead of the method of color compression.

- In addition, there are command line tools that can be entered to the DOS prompt, such as BMP2TIM.EXE (from Windows BMP to TIM), PICT2TIM.EXE (from Macintosh PICT to TIM), and RGB2TIM.EXE (from general-purpose RGB format to TIM). You can utilize them when you are not using Windows, or when you perform the same conversion repeatedly. For use of each tool, see the appropriate reference manual.

CHAPTER 4

EDITING OF MATERIAL DATA

This chapter describes how to edit material data using Material Editor.

1 What is Material Editor

Material Editor is a tool that edits surface attributes (materials) of polygons, such as colors, textures (patterns), and method of shading of 3D models.

By setting various surface attributes, you can make the model look real. Material Editor allows setting of the following surface attributes, which can be expressed on PlayStation, to each polygon:

- Color
- Texture
- Transparency (opaqueness, or semi-transparency)
- On/off of light source calculation
- Shading method (flat or smooth)

Tutorial 1 in this chapter describes fundamental operations, and then Tutorial 2 explains advanced functions.

2 Tutorial 1 (Fundamental Operations)

This tutorial introduces fundamental use of Material Editor using a sample model (body of the helicopter). If Material Editor and ARtist Board are not installed into your machine, install them, referring to “INSTALLATION” CHAPTER in Reference Manual.

[Starting Material Editor]

Let us start Material Editor.

(1) To Start Material Editor, execute MEDITOR.EXE under the standard directory (C:¥PSXGRAPH¥BIN) for the 3D graphic tools. If the program is saved to Program Manager for Windows, you can start Material Editor using Program Manager by double-clicking the following icon marked with “Material Editor”:

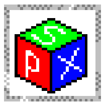


Figure 4-2-1 Icon in Material Editor

When Material Editor has been started, the opening window displaying the Material Editor version number and copyright pops up.

Figure 4-2-2 Opening Window

Remarks:

If the version number of your Material Editor is earlier than 1.70 (e.g. 1.68c), install the latest version from floppy disks or BBS. This tutorial is intended for Material Editor version 1.70 or later. Normal operation is not guaranteed if earlier version is used.

Remarks:

Material Editor uses Graphics Artist Board (DTL-H201A). Upon starting, the address written in the Artist Board port address setting file (C:\WINDOWS\ABOARD.INI) is used as the port address of Artist Board. For instance, if ABOARD.INI specifies

“addr=0x1340”,

Material Editor communicates with Artist Board using the port address 0x1340. Normal communication cannot be performed unless the port address of the board is set to 0x1340 using DIP switches. If the DIP switches are not properly set, the following warning window will appear, and Material Editor will be terminated.

Figure 4-2-3 Port Address Warning Address

In this case, correct the ABOARD.INI file using the ABOARD.EXE tool, or change the address of the board itself so that the values of these addresses may become the same.

For the ABOARD,EXE tool, see 5.1, “ABOARD.EXE”, in PART III, “REFERENCE”. For the procedure for changing the port address of Artist Board, see the manual for Artist Board.

(2) Click the “OK” button on the opening window, and main window of Material Editor appears.

Figure 4-2-4 Main Window of Material Editor

[Loading model data]

Let us load model data.

(1) Select “Open” command on file menu.

(2) A file selection dialogue box appears on the screen. Select the following file, and clock the “OK” button:

C:¥PSXGRAPH¥TUTORIAL¥MEDITOR¥TUTO1¥RSD¥HELI.RSD

(3) The "Texture layout" dialogue appears on the screen. Without minding what the dialogue box tells you, click the "OK" button.

A helicopter is displayed on the video monitor. It is a finished helicopter which you create in Tutorial 1.

Figure 4-2-5 Finished Helicopter

[Rotation of Helicopter; Parallel displacement of viewpoint]

While rotating the helicopter, let us see it from various angles.

(1) The following keys allow you to rotate the helicopter:

A	Left	S	Twisted clockwise
D	Right	C	Twisted counterclockwise
W	Up		
X	Down		

(2) You can move the viewpoint parallel.

Shift-A	Left	Shift-S	Inward
Shift-D	Right	Shift-C	Toward front
Shift-W	Up		
Shift-X	Down		

In either case, holding down the key continues to rotate or move the model.

You can find the above key binds any time if you select the “key bind” command.

When editing materials, you have to frequently rotate a model, or move it parallel. So, you should become familiar with model operation from the keyboard.

Remarks:

The helicopter will not move unless Material Editor is selected. Before manipulating the keys, be sure to select main window of Material Editor.

Tips:

If you shorten “auto repeat start time” and “repeat interval” for the keyboard, the object will be rotated and moved smoothly. You can change these settings by double-clicking “control panel” in Program Manager for Windows to start the “keyboard”.

[Loading helicopter data]

Let us load data to which no material is set. Following the same steps you followed earlier, call up the following file:

C:¥PSXGRAPH¥TUTORIAL¥MEDITOR¥TUTO1¥RSD¥HELI00.RSD

Now, a white helicopter appears on the video monitor. It is a DXF file created with a commercially available modeler converted into RSD using the DXF2RSD converter.

Figure 4-2-6 Initial Status of Helicopter

[Painting the canopy]

Let us change the color of the canopy of the white helicopter cockpit to light blue.

1) Click material menu, and the following “material setting” dialogue appears:

Figure 4-2-7 Material Setting Dialogue

This dialogue box has many radio buttons and push buttons used to set materials. Do not push any button yet.

The top part of the dialogue box shows the RSD file currently loaded, and number of polygons contained in the object. Below these items, you see ID of the polygon you selected last.

2) Click the “polygon selection” button at the left bottom of the material setting dialogue box.

The mouse cursor on the Windows screen disappears, and a white cross cursor appears on the video monitor. (If you cannot find the cross cursor, move the mouse to see it.)

This status is called “polygon selection mode”. In this mode, one polygon on which the cross cursor is selected if you left-click the mouse. If there are two or more polygons under the cross cursor, the foremost one is selected. You cannot select a polygon whose back is faced up.

You can easily identify a selected polygon because it turns red, and the boundaries are displayed in yellow.

To cancel polygon selection mode, you may right-click the mouse. Then, the mouse cursor will return to the Windows screen.

3) Moving the cross cursor and helicopter, select all the polygons within the canopy.

As shown in the figure below, you select sixteen polygons in all.

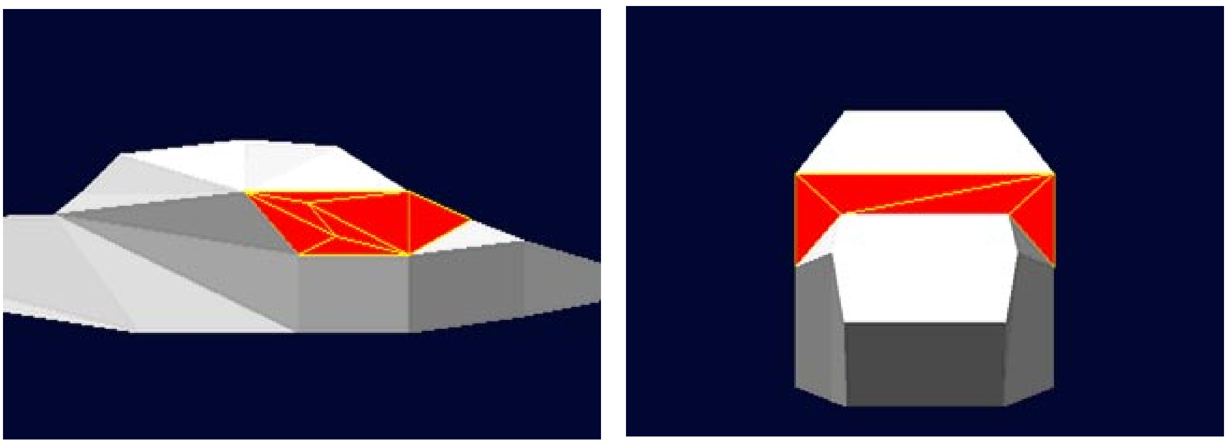


Figure 4-2-8 Selection of Canopy

To cancel selection of a polygon, click that polygon again while holding down the control key.

When the polygons within the canopy have been selected, right-click the mouse to quit from polygon selection mode.

Tips:

To select all the polygons within the canopy, you have to rotate the helicopter, and move it to and fro, and to right and left. You can do so easily if you select any desired polygon by manipulating the cross cursor with the right hand while moving the model by operating the keys with the left hand.

4) Click the “color” radio button on the material setting dialogue box.

Windows standard “color setting” dialogue box appears. Select a color suitable for the canopy. The color setting dialogue box is a Windows standard function. For details, see the manual for Windows.

Click the OK button on the color setting dialogue box, the selected polygon is painted with the selected color.

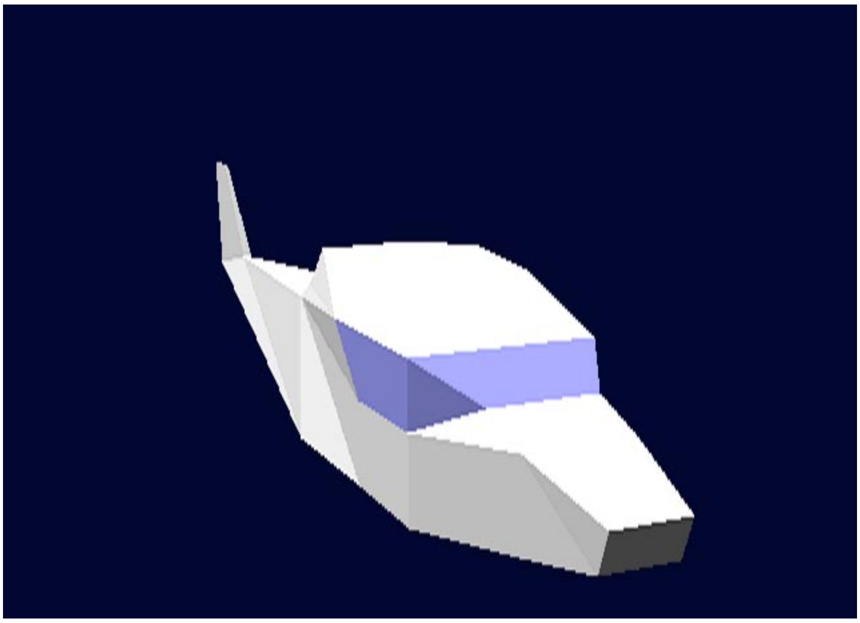


Figure 4-2-9 Canopy Painted in Light Blue

Tips:

If you do not like the painted color, you can paint the canopy in any desired color as often as you like by clicking the color radio button without selecting the polygon again if you have not pressed the OK button in the color setting dialogue box.

[Saving the canopy as a group]

Let us see the group function of Material Editor. A group is a set of polygons. You can create a group freely. You can select all the polygons belong to a group by selecting the group. For instance, you save polygons to be painted in the same color as a group, you can select all these polygons at one time whenever you want to change their color.

Let us save the canopy (sixteen polygons) as a group. If you have not painted them, select the "open" command on the file menu to load the following file, which contains sample data:

C:¥PSXGRAPH¥TUTORIAL¥MEDITOR¥TUTO1¥RSD¥HELI01.RSD

1) Click the "polygon selection" button on the material dialogue box to select the sixteen polygons for the canopy.

2) Right-click the mouse to cancel polygon selection mode, and select group menu.

   Then, a group control dialogue box appears on the screen.
   You will find that the "new group" part at the top indicates "16" as the number of selected polygons.

3) Click the "save" button on the group control dialogue box.

4) A group name dialogue box appears. From the keyboard, enter "canopy" as the group name.

5) Click the "OK" button on the group name dialogue box. The "canopy" group is now saved. The group name and the number of polygons are displayed in the group list.

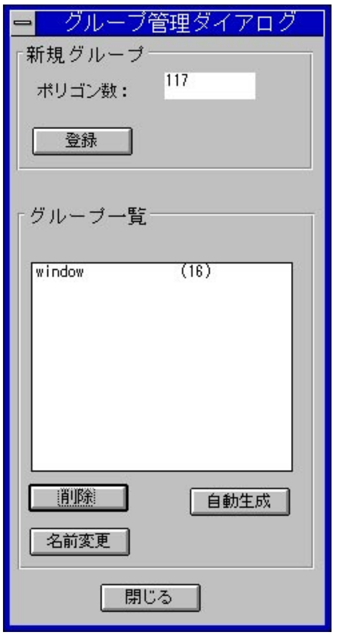


Figure 4-2-10 Group Control Dialogue Box Saving Canopy Group

6) Click the line indicating “window(16)” in the group list. On the video monitor, make sure that all the polygons for the canopy are selected at one time.

7) While holding down the control key, select “window(16)”. On the video monitor, you will see selection of polygons for the canopy is cancelled.

8) After selecting “window(16)”, press the “cancel” button. The window group is cancelled, and deleted from the list.

9) Press the “automatic creation” button.

The automatic creation function automatically creates a group according to the type of material. Because the helicopter has the two types of materials, or the white part and light blue part (canopy), two groups are created by the automatic creation function.

10) Select “Group_0” from the group list. Then, press “Group_1” while holding down the control key.

You can select the two groups. In this manner, you can select any desired number of groups from the group list at one time.

[Making the canopy semi-transparent]

Let us make the canopy semi-transparent so that it may look like a windowpane.

1) Select the canopy group on the group control dialogue box.

2) Click the “semi-transparent” radio button on the material setting dialogue box.

Now, the canopy is made transparent, and you can see the interior of the cockpit.

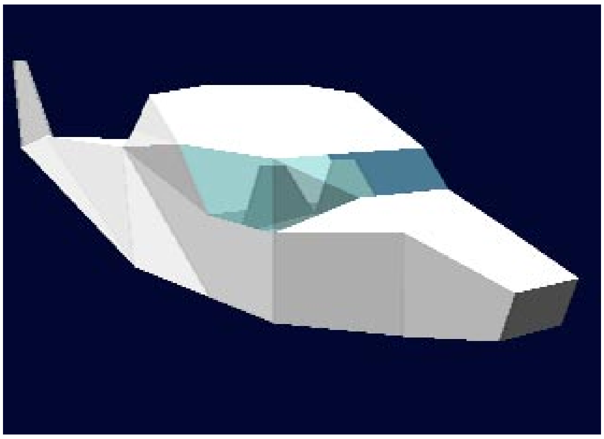


Figure 4-2-11 Semi-Transparent Canopy

[Painting the cockpit]

Now, you can see the interior. Let us paint the cockpit. To select polygons (hidden polygons) surrounded with other polygons, such as the cockpit, either of the following two methods is used:

Method 1)

1) Move the cursor onto the cockpit across the canopy.

2) If you click the mouse, the canopy is selected. Click the mouse without moving it, click it again. Then, the polygon for the cockpit which located behind the canopy is selected.

Each time you click the mouse without moving it, a polygon behind the currently selected polygon is selected. This method is effective to select polygons behind transparent polygons as in the case of the canopy.

Method 2)

1) Select a polygon that hides the polygon you want to select.

2) Using the “display off” radio button on the material setting dialogue box, temporarily turn off the display of these polygons.

3) Then, you can easily select any desired polygon because the obstructive polygons are removed.

In either case, you may save the hidden polygons as a group whenever you select them. If you do so, all you have to do thereafter will be to select that group.

[Pasting a texture to the body]

Let us paste a texture to the body.

1) Load the following file as sample data:

C:¥PSXGRAPH¥TUTORIAL¥MEDITOR¥TUTO1¥RSD¥HELI02.RSD

2) Press the F2 key to select “parallel projection” mode. Parallel projection is helpful to determine the texture position precisely. If you press the F2 key again, fluoroscopic projection mode will be selected.

3) Click the “texture” radio button on the material setting dialogue box.

The file dialogue box opens. So, select the HELISIDE.TIM file, and click the “OK” button.

Now, the texture layout dialogue box appears. This dialogue box allows you to specify how the selected texture should be located on VRAM. Information (filename, pixel size, and mode) related to texture data is also displayed.

4) Enter the following to the texture layout dialogue box:

Texture X: 640
Texture Y: 0

CLUT X: 0
CLUT Y: 480

Then, click the “OK” button on the texture layout dialogue box.

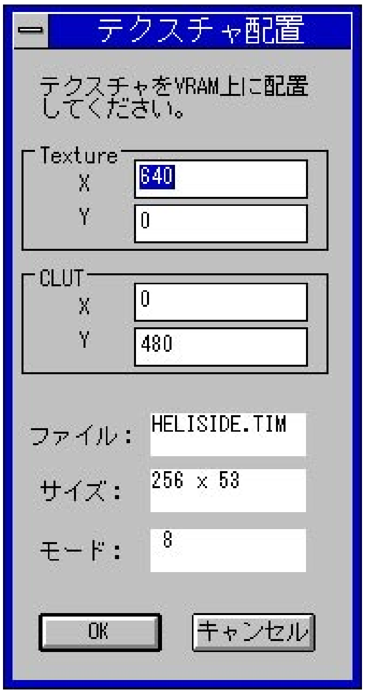


Figure 4-2-12 Texture Layout Dialogue Box

Now, texture data in the TIM file named HELISIDE.TIM is transferred to VRAM on Artist Board.

The texture mapping dialogue box appears on Windows. The semi-transparent texture of the helicopter is also displayed on the video monitor. This semi-transparent texture is the source texture you are going to map.

Tips:

CHAPTER 3, “CREATION OF TEXTURE DATA”, in PART II, describes how to specify addresses for texture layout.

5) The texture will be projected parallel to the side of the helicopter. So, rotate the helicopter so that its side may be faced to you, and the head of the helicopter may be faced to the right side of the screen.

Tips:

Each time you press the “Ctrl+A” key combination, the helicopter is rotated by 90 degrees. You can face the side of the helicopter toward you by pressing the “Ctrl-A” keys several times after the data is loaded.

6) Select the heliBody group on the group control dialogue box.

   Thus, all the polygons to which you will paste textures are selected.

7) Press the “auto” button on the texture map dialogue box. The position and size of the texture is automatically changed so that the source texture may be precisely put on the selected polygon.

Polygons which were displayed in red turns yellow. Each polygon present either of the following states:

Red polygon: Selected, but not included with the source texture
Yellow polygon: Selected, and included with the source texture

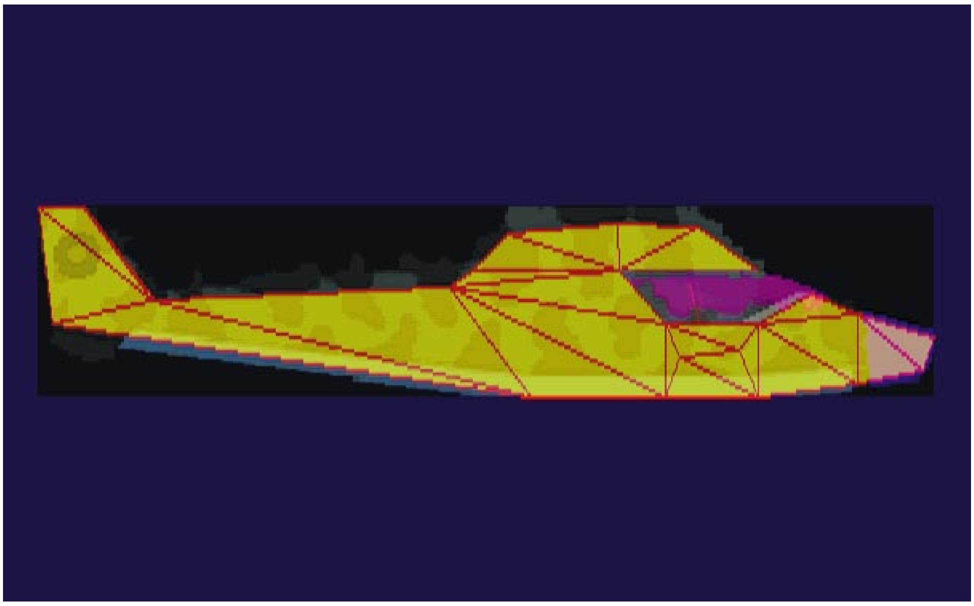


Figure 4-2-13 Automatically Adjusted Source Texture

8) Click the “OK” button on the texture dialogue box. You can paste the texture to the helicopter body.

TIPS:

Depending on the texture and shape of the model, you may have to adjust the position and size of texture manually. You can move and re-size the source texture using the following keys:

←	Move to left	→	Move to right
↑	Move up	↓	Move down
Ctrl- ←	Horizontal shrinkage	Ctrl- →	Horizontal enlargement
Ctrl- ↑	Vertical shrinkage	Ctrl- ↓	Vertical enlargement

Figure 4-2-13 Handling of Texture from Keyboard

9) Check the effect of the texture map by observing the helicopter from various angles. The textures at both sides of the helicopter must have a protective color.

However, the top and bottom of the texture are distorted, and they are not painted in a protective color. In parallel projection mapping of a texture, the mapped texture is not distorted if the target polygon is perpendicular to the mapping direction. As the mapping direction is deviated, distortion of the texture becomes more outstanding.

To give a protective color to both the top and bottom of the helicopter, you have to prepare textures for the top and bottom separately, and map them to the top and bottom perpendicularly. This tutorial omits these steps.

[Making the body smooth]

Lastly, let us make the body of the helicopter smooth. The following file contains sample data. Load it by selecting the “open” command on file menu.

C:¥PSXGRAPH¥TUTORIAL¥MEDITOR¥TUTO1¥RSD¥HELI04.RSD

1) Select “heliBody” on the group control dialogue box. The helicopter body turns red.

2) Click the “smooth” radio button on the material setting dial. Then, the boundaries of polygons for the helicopter body become less outstanding.

REMARKS:

Once polygon selection mode is selected, the smooth-shaded portions are flat-shaded. To display smooth-shaded polygons, click the “polygon cancellation” button.

This finishes Tutorial 1.

3 Tutorial 2 (Applied Use)

This tutorial explains the following functions, which were not discussed in Tutorial 1.

- Snap shot
- Decal
- Editing and loading of vertexes
- Plunger and brush functions

First, load the following file into Material Editor:

C:¥PSXGRAPH¥TUTORIAL¥MEDITOR¥TUTO1¥RSD¥HELI20.RSD

[Snap shot]

Snap shot is the function to save the image of the video monitor to a TIM file. You can create texture data easily by editing the TIM file using PhotoShop, which is equipped with the TIM plug-in.

For instance, you may load the helicopter to which no texture is pasted to Material Editor, and take a snap shot with the side of the helicopter faced to you. Then, apply a protective color to the side view of the helicopter in the TIM file. If you use this TIM file as texture, you can precisely fit a texture to the side of the helicopter.

1) Select the “snap shot” command on file menu.

A file dialogue box appears. Enter a filename (FOO.TIM) under which you want to save the file.

2) Using TIMUTIL.EXE or PhotoShop (with TIM plug-in), display FO.TIM on Windows screen.

[Decal function]

Load HELI20.RSD and HELI21.RSD to Material Editor, and compare them with each other. Letters “PS” and “1995” are texture-mapped on the side of HELI21.RSD. These letters were not written to the texture image having the protective color. A separate texture having letters were put on,and pasted to, the texture having the protective color.

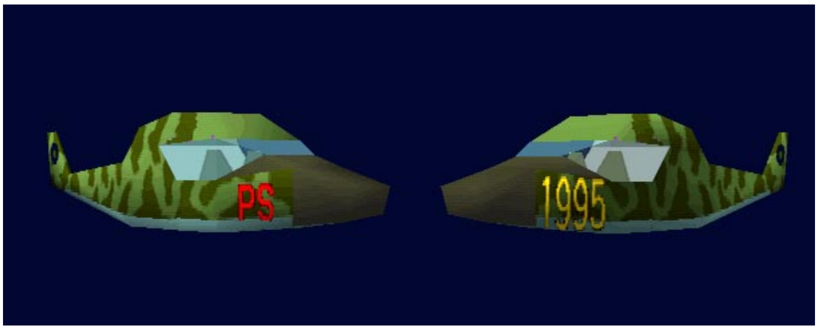


Figure 4-3-1 Helicopter to which Decal is Pasted

Do realize the decal function on PlayStation, do as follows:

1) Paste the base texture (protective color).

2) Prepare double polygons to which a decal is to be pasted.

To obtain double polygons, select a polygon, and click the “copy” button on the material setting dialogue box.

3) As a letter texture, prepare a texture which is transparent (STP = 0, (R, G, B)= (0, 0, 0), excepting the letters.

4) Paste the above-mentioned letter texture to the front side polygon between the double polygons.

TIPS:

There is no function which deletes a copied polygon. In actual data creation, it is advisable to save the model before copying it.

TIPS:

If you click the double polygons, the one at the front side is selected. If you click them twice without moving the cursor, the inner one is selected.

If the outer polygon is selected, it looks red. If the inner polygon is selected, it does not look red because it is hidden by the outer polygon. However, if you see it carefully, the edges of the inner polygon look yellow.

In the above example, decal function is used to paste textures. If you paste a texture decal to a color polygon, there will be no difference in colors of the base color of the texture portion and polygons surrounding it.

Figure 4-3-3 shows an example in which the same color is applied to the base of the letters and surrounding polygons without using the decal function. Figure 4-3-3 is an example in which the decal function is used.

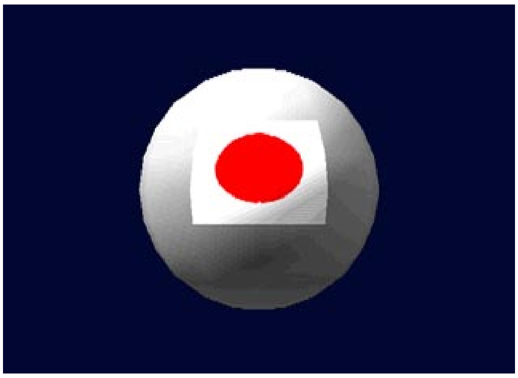


Figure 4-3-2 Decal Pasted to Color Polygon

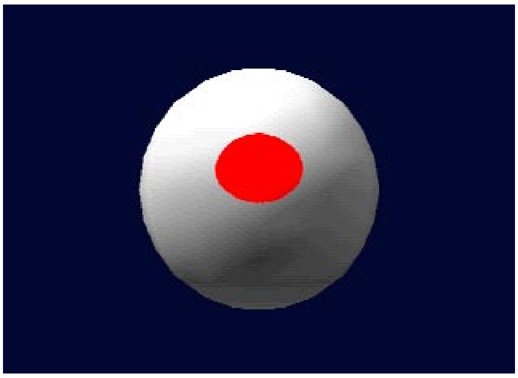


Figure 4-3-3 Use of Decal

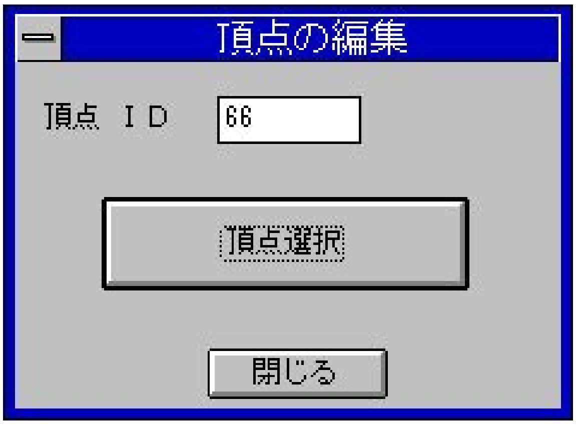
[Editing and loading vertexes]

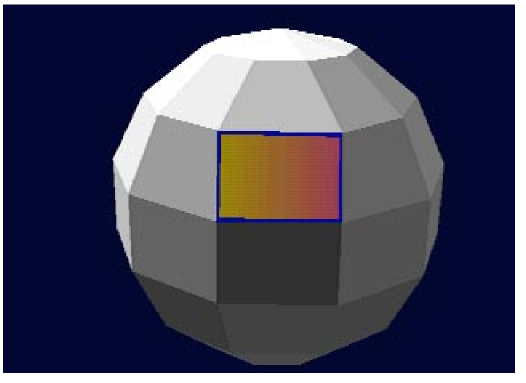
"Editing of vertexes" is a tool which moves 3D model vertexes parallel. Normally, it is used for fine adjustment of the shape of a model. If you use this function, you can fine-adjust UV values for texture mapping.

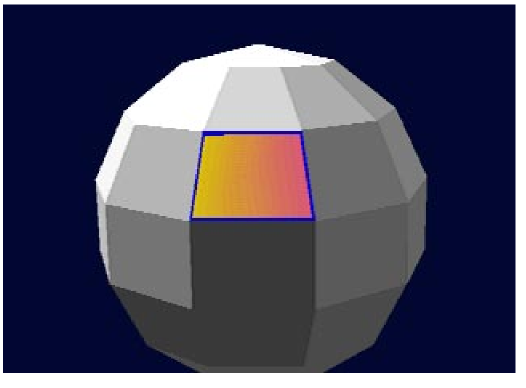
For your reference, load the following file:

C:¥PSXGRAPH¥TUTORIAL¥MEDITOR¥TUTO1¥RSD¥BALL00.RSD

Three textures (one rectangular polygon, and two trapezoidal polygons) are pasted to this model. You can quickly perform texture mapping of the rectangular polygon if you use the automatic fine adjustment function on the texture map dialogue box. However, if you use the automatic fine adjustment function for the trapezoidal polygons, one corner of the texture will extend from the polygon, and the blue boundary cannot be precisely put onto the boundary of the polygon.


頂点の編集
頂点 ＩＤ
66
頂点選択
閉じる





CHAPTER 5

DISPLAY ON PLAYSTATION

This chapter describes the procedure for displaying 3D model data on PlayStation.

So far, we displayed 3D model data using Material Data through Artist Board. In this chapter, and chapters that follow, we will describe how to create data which can be displayed on PlayStation. The format for this purpose is TMD. This format is directly read by the library, and loaded to CD-ROM. It is the format to be used by the programmer. However, we think it is necessary for an artist to check the final output.

TMD can be obtained by converting RSD using the 3D graphic tool.

To display TMD, “PlayStation Board” and “Programming Tool” are needed. Ask a person in charge of the entire program for cooperation, and perform the following practices.

1 Creation of TMD

When RSD data has been created, it can be converted into TMD by executing RSDLINK.EXE. TMD is a format suitable for execution of a game on PlayStation. Generally, it is directly loaded to CD-ROM. RSDLINK.EXE forms multiple RSD files into one file. Sample data is saved to the RSD and TIM subdirectory under the HELIVIEW directory. First, move the current directory to the RSD directory by entering the following to the DOS prompt:

```
> CD HELIVIEW¥RSD
```

RSD.LINK is generally executed under the RSD directory.

Give the following RSDs, which are drawn within the same frame, to the arguments of RSDLINK.EXE. In this example, two arguments ar used. If there are a large number of arguments, it is helpful to save them to a .ARG file. The use when they are formed into a file named HELI.ARG is also shown below.

```
> RSDLINK     HILI ROTOR
```

Otherwise,

```
> TYPE        HELI.ARG
  HELI
  ROTOR
> RSDLINK     HELI.ARG
```

As a result a file named A.TMD is created. Thus, the data can be displayed on PlayStation.

2 Options of RSDLINK.EXE

The fundamental options of RSDLINK.EXE are explained below.

Designation of output filename

-o outfile-name

Specifies the output TMD filename.

Enlargement/shrinkage

-s factor

Generates a TMD file by expanding (shrinking) RSD data by the scale factor (rounded to the power of 2) given by the argument. Therefore, the TMD file thus created has the different scale from the original RSD data. This option is helpful when the scale of RSD is too small. However, it is recommended to basically perform resizing using RSDFORM.EXE, which will be described in 5.5.

Creation of object ID table

-id model.prj

Reads a project file, and creates an ID header file. It is output as a header file in C language, and used by an animation program. (See CHAPTER 6.)

For other options, see 2.9, "RSDLINK.EXE", in PART III.

3 Display of Helicopter

Let us display the TMD file on PlayStation. To do so, you must use a program which loads TMD and draw it. As a sample program, use a program (such as "TMDVIEW5"), which displays TMD. Compile it, and load the TMD file and TIM file for the helicopter, and execute. You can move the helicopter using the controller connected to the PlayStation board.

4 Adjustment of Model

You will sometimes find the size too small, or the position is improper when you display TMD. You may see nothing on the screen because the size is too small. The cause is the too small scale for DXF data when you originally created the model using the 3D modeler. You did not notice it because Material Editor displays an object to the optimum size and in the best-suited position. In such a case, re-size or move the object using RSDFORM.EXE. You can make adjustments after pasting of textures without returning to the 3D modeler.

In addition to parallel translation, RSDFOR.EXE can perform enlargement/shrinkage and rotation for each axis independently. You can create a shape (mirror image) in a symmetric position. Using this, you can create similar derivative models from one model.

5 Options of RSDFORM.EXE

Designation of output filename

-o output RSD name

Specify the output RSD name. If you fails to do so, the RSD name will be “a.rsd, a.ply,...”. The file will not be overwritten.

Parallel translation

-t x y z

This optional switch moves the model parallel in the directions of the X-, Y-, and Z-axes for a distance given by the argument.

Enlargement/shrinkage

-s x y z

This option enlarges or shrinks the model in the directions of the X-, Y-< and Z-axes by the scale factor given by the argument. If a negative value s specified, a mirror image with respect to the axis concerned is created.

Information display

-v

This option displays the coordinates of the maximum/minimum and center of each axis before and after modification. When you want to precisely move the center of gravity to a certain position, you can confirm the coordinates using this option.

For other options, refer to Section 2.8, “RSDFORM.EXE”, in PART III, “REFERENCE”.

If the helicopter in the sample program fails to be displayed properly, make fine adjustment using RSDFORM.EXE.

To understand applied use, let us change the shape of the helicopter using the enlargement/shrinkage function.

```
> RSDFORM -o HELI9 -s 1 0.5 1 HELI8.RSD
```

If you enter the following, a vertically crushed helicopter (HELI9.RSD) will be vertically crushed:

```
> RSDFORM -o HELI9 -s 10.51 HELI8.RSD
```

If you enter the following, a transversely crushed helicopter (HELI10.RSD) will be created. Because the model is rotated (by _1 turn) in the direction of the Y-axis, the front and rear are reversed.

Using RSDFORM.EXE, try to create various helicopters.

CHAPTER 6

ANIMATION TOOL

This chapter describes the following fundamental operations of Animation Tool (Ver.1.x):

- Layout of a model
- Operation of the camera
- Import of a DXF file
- Creation of a hierarchical structure
- Creation of an animation

For this tool, you can use only the left-hand button of the mouse. If you want to quit from the tool, you may select "Quit" on file menu.

1 Basic Operation of Model

Displaying a model

Start animatio.exe.
Select "Open Project" command on project menu. The following dialogue box appears.

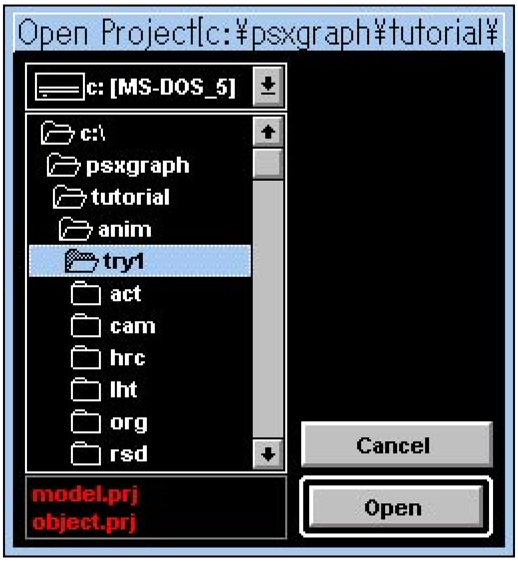
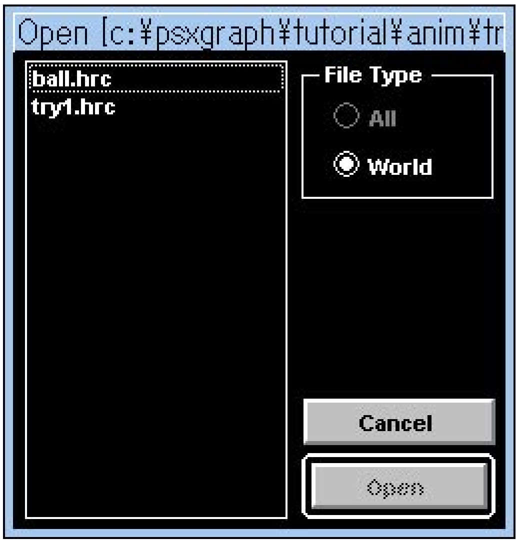


Figure 6-1-1 Opening Project

Double-click the directory field to move the current directory to C:¥PSXGRAPH ¥TUTORIAL¥ANIM¥TRY1. When the suitable directory is selected, you can manipulate the "Open" button. A this time, red letters "model.prj, object.prj" are displayed. They indicate that the current directory is the Project directory. Do not select it. Click the "Open" button, and the dialogue box disappears. Thus, the project has been opened. Now, the directory where you will perform work is determined. In this status, no model is displayed on the screen.

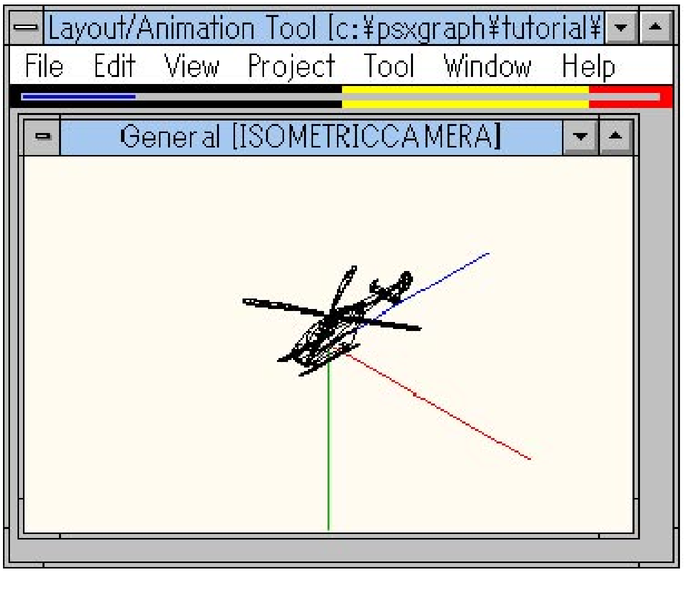
Now, select the “Open” command on file menu. The following dialogue box appears:

Figure 6-1-2 Opening HRC



Select try.hrc displayed in the file field. When the file is selected, you can use the “Open” button. Click the “Open” button. Then, the dialogue box disappears, and the general window appears. Thus, the HRC file has been opened. Do you see a helicopter? The red, blue, and green lines indicate the positive directions of the X-, Y-, and Z-axes of the world coordinate system.

Figure 6-1-3 Open HRC (Helicopter)



This helicopter is used in the following description. When you quit from the tool, do not save the HRC file.

Creating a model

This section descries the procedure for model selection, parallel translation, rotation, and enlargement/shrinkage.

1) Selecting a model

Put the mouse cursor on the helicopter, and click it. A red rectangular parallelopiped covering the helicopter appears. It is called a “bounding box”. This finishes the selection of an object using the mouse.

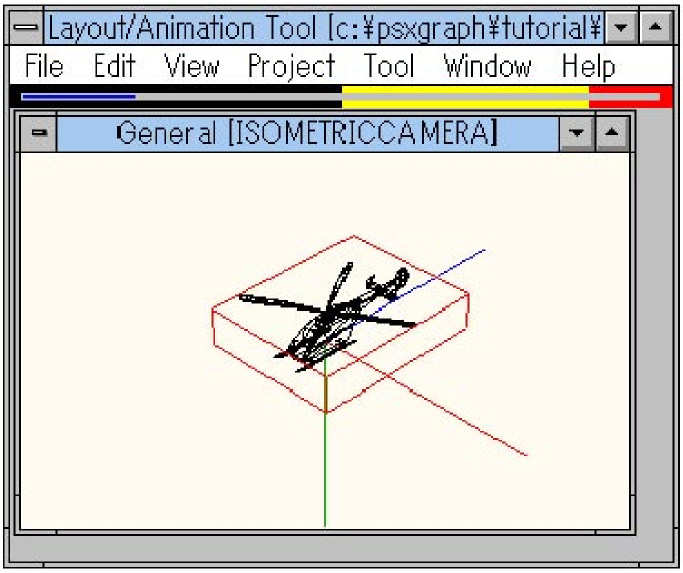


Figure 6-1-4 Selected Helicopter

2) Parallel translation

Put the mouse cursor on the bounding box, and drag the mouse cursor. According to movement of the mouse, the bounding box moves parallel. Release the mouse button when the bounding box is moved to a suitable position. The helicopter is displayed in the position of the bounding box.

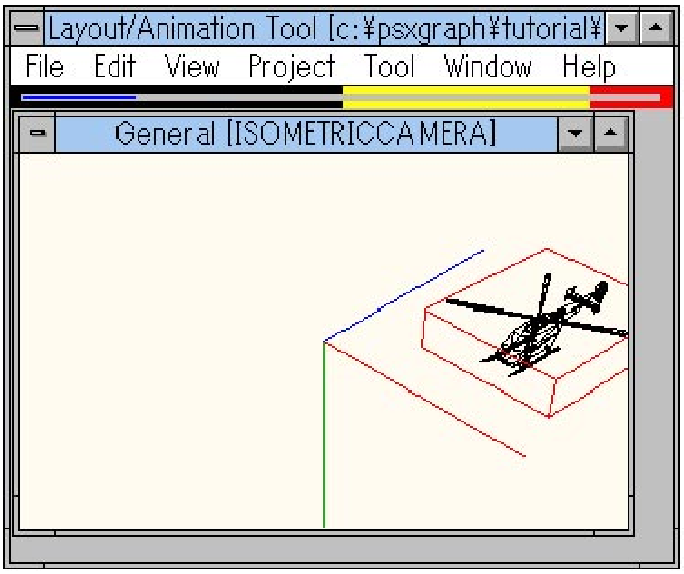


Figure 6-1-5 Parallel Translation

While holding down the Shift key, put the mouse cursor on the bounding box, and drag the mouse toward the front. You will see the bounding box moving toward the front. If you want to move it backward, drag the mouse backward. This is the parallel translation of the object. Strictly speaking, the object is moved in parallel with the viewpoint coordinate system. Therefore, when an object which is not located at the center of the screen is moved to the front or rear, it looks as if it is moving along a radiant line drawn from the center of the screen while its size is changed.

3) Rotation

Put the mouse cursor to a vertex of the bounding box, and drag the cursor. The bounding box rotates as the mouse moves. Release the mouse button. The helicopter is displayed in the same posture as the bounding box. This method may be used to adjust the posture roughly.

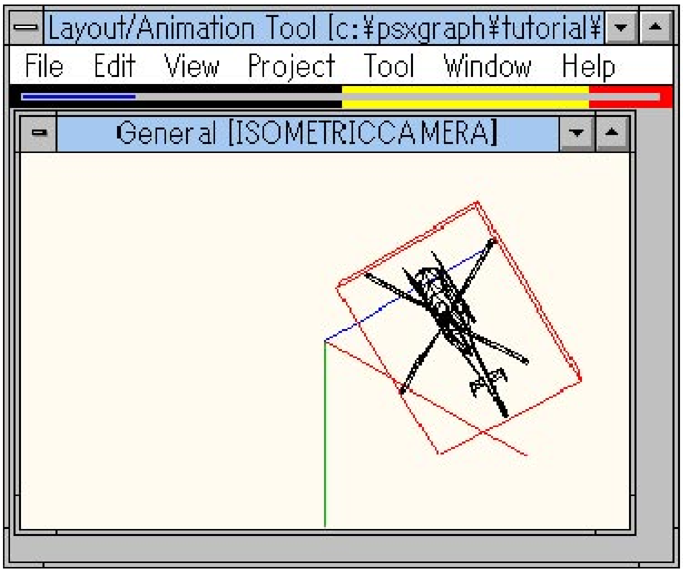


Figure 6-1-6 Rotation by selection of a vertex

Then, put the mouse onto a line of the bounding box, and drag it. The bounding box rotates around an axis parallel with the line. Release the mouse button. The helicopter is displayed in the same posture as the bounding box. This method may be used to adjust the posture.

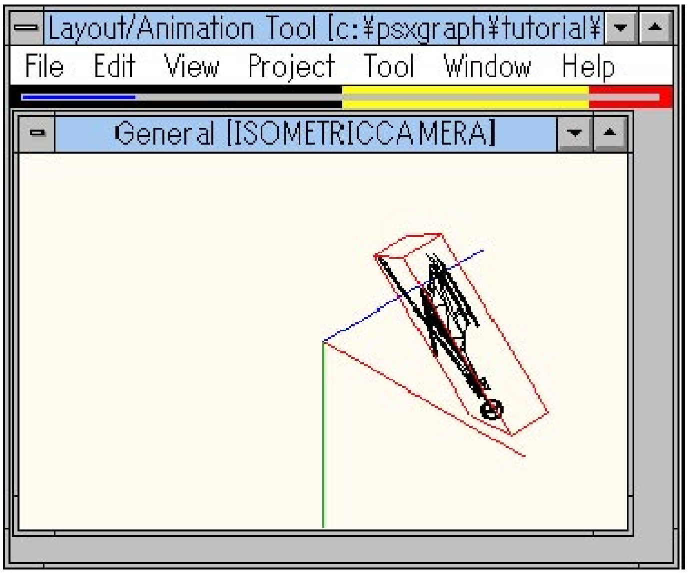


Figure 6-1-7 Rotation by selection of a ridgeline

This is the rotation of the object. The center of rotation is always the center of the bounding box. In the first method, the direction of the rotation axis always changes. In the second mouse, it is constant until you release the mouse button.

4) Enlargement/shrinkage

While holding down the Ctrl key, put the mouse cursor onto the vertex of the bounding box, and drag mouse so that it may go farther from the center of the bounding box. The bounding box is enlarged as the mouse is moved. When the size is almost doubled, release the mouse button. The helicopter of the same size as the bounding box is displayed. If you want to shrink, drag the mouse so that it may be brought near to the center of the bounding box.

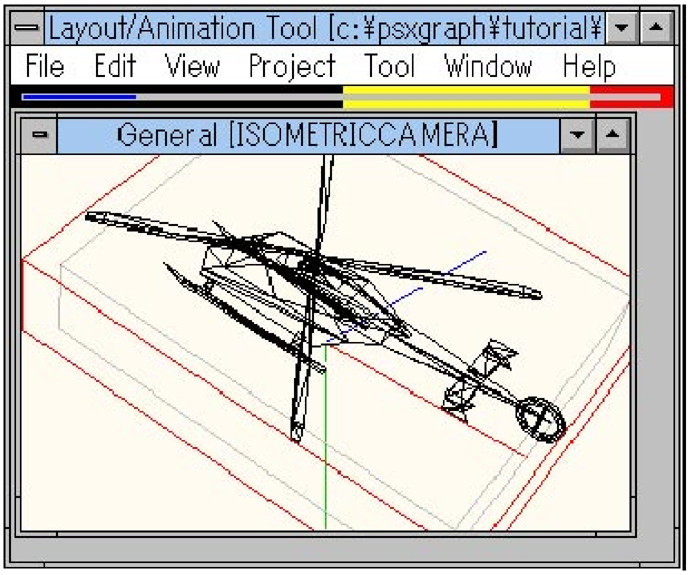


Figure 6-1-8 Enlargement by Selection of Vertex

While holding down the Ctrl key, put the mouse cursor to a plane of the bounding box, and drag the bounding box. If you drag the plane so that is may go far from the bounding box, the plane is drawn out and enlarged. If you bring it near to the center, the object shrinks as if it were crushed by that plane.

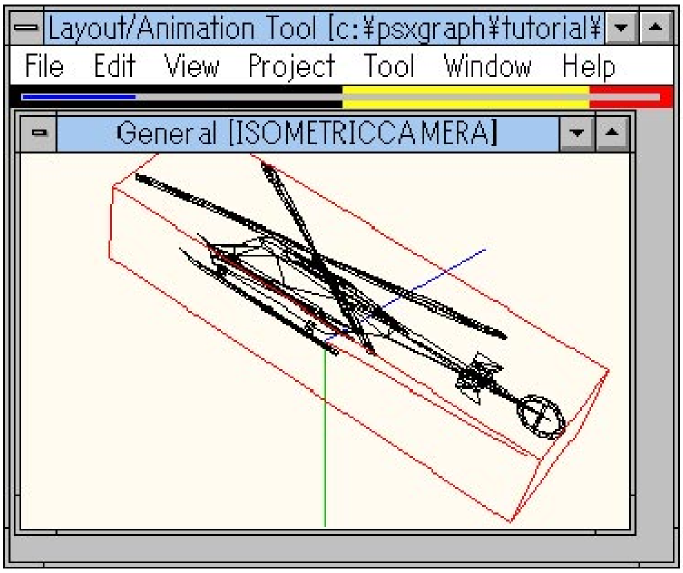


Figure 6-1-9 Enlargement/Shrinkage by Selection of a Plane

This finishes enlargement and shrinkage of an object. The first method is proportional enlargement and shrinkage not dependent on the direction. The second method is enlargement and shrinkage having a direction. The scale factor for enlargement and shrinkage is 1/4096 time to about 8 times (8 times to 1/4096 time) because of the restrictions imposed by PlayStation.

These are simple functions of the tool. Please understand that data are manipulated inside the computer with ease with which you hold a cup.

5) Manipulation by entry of numerical values

Let us see how to manipulate an object by specifying numerical values, such as rotation angle and amount of parallel translation. Close the file in the status 4), and open the helicopter again.

When it is opened, call up the tool palette by selecting the “Tool Palette” command on the tool menu.

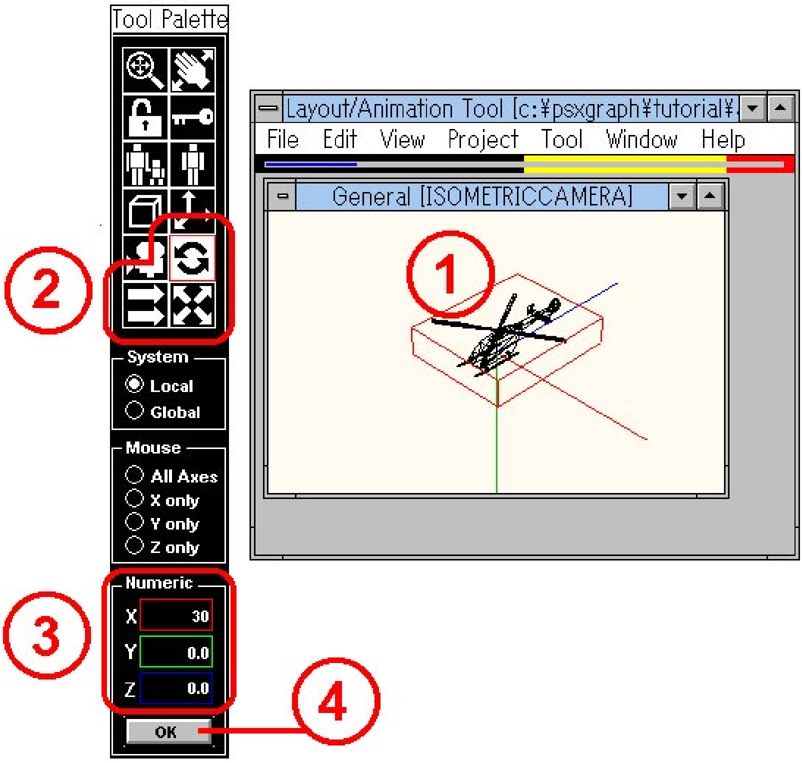


Figure 6-1-10 Manipulation by Entry of Numerical Values

To enter numerical values, follow these steps:

① Select an object.

② Specify the method of operation.

③ Enter numerical values.

④ Click the OK button.

① and ② may be performed in the reverse order. In step ③ parallel translation is performed using the current position as starting point. A negative value can also be specified. For rotation, you specify an angle. The position of the object is changed from the current posture. You can also use a negative value. For enlargement and shrinkage, 1.0 does not change the size. You must not specify a negative value.

2 Fundamental Operation of Camera

The image displayed on the window is relayed through an imaginary camera. You will see “General [ISOMETRIC CAMERA]” on the title bar on the window. It indicates that an image as seen from the isometric camera is displayed on the general window. This section describes how to change screen display on the window by pushing, pulling, swinging up and down, to right and left, and tilting the imaginary camera. As in the previous section, we will use the helicopter. Open the HRC file. If the helicopter is not put to the center of the screen, move it by following the procedure described in the previous section.

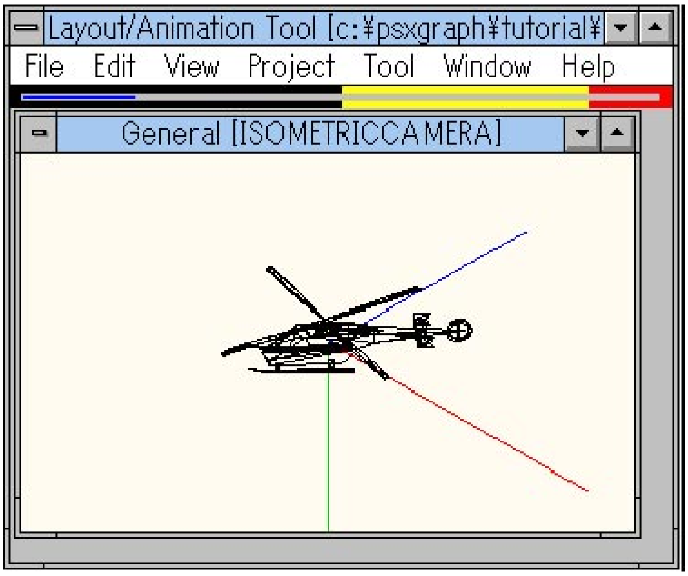


Figure 6-2-1 Fundamental Operation of Camera

Pushing and pulling the camera

To bring the camera near the helicopter, strike the space bar. Each time you strike it, the camera moves forward for 1000. If the camera goes too far, and passes by the helicopter, nothing will be displayed. To move back the camera, strike the space bar while holding down the Shift key. Each time you strike the space bar, the camera moves back for 1000. If you move back the camera to a ceratin point, you will see the helicopter. Holding down the space bar does not move the camera forward or backward.

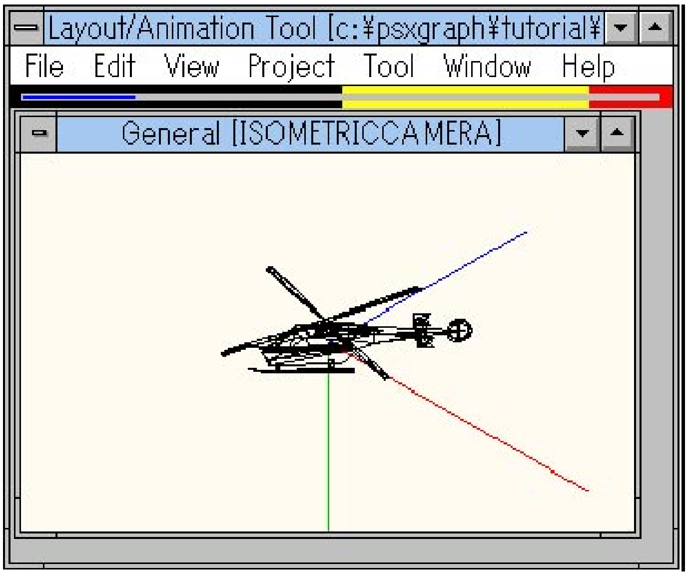


Figure 6-2-2 Pushing/Pulling the Camera (Large)

Move the camera to a position where it just passes by the helicopter. To move the camera close to the helicopter from this position, strike the space bar while holding down the Ctrl key. Each time you strike it, the camera approaches for 100. To move the camera backward, strike the space while holding down the Shift and Ctrl keys. Each time you strike the space bar, the camera is moved backward for 100.

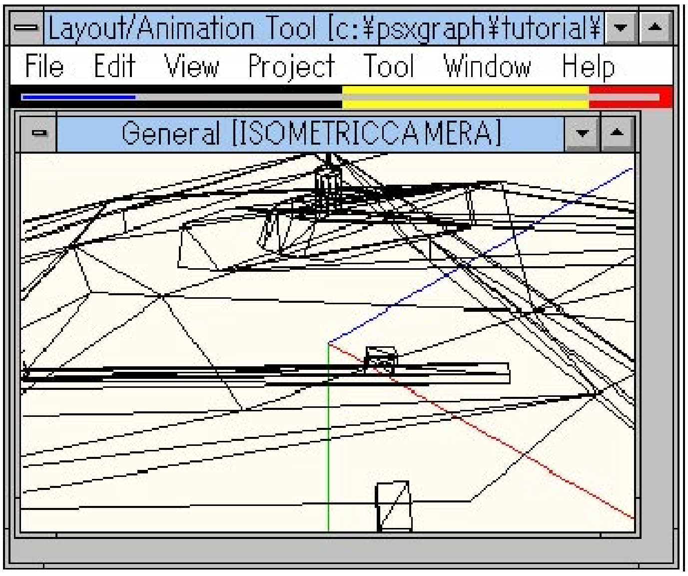


Figure 6-2-3 Pushing/Pulling Camera (Small)

Swinging the camera up and down, and to right and left

The direction of the camera is determined by the cursor keys. Strike the right cursor key. Now, you will see the right portion of the helicopter. To change the direction to the left, strike the left cursor keys. Do the same to move the camera up and down. Each time you press the cursor key, the direction of the camera changes by 10 degrees. The direction will not change even if you hold down the cursor key.

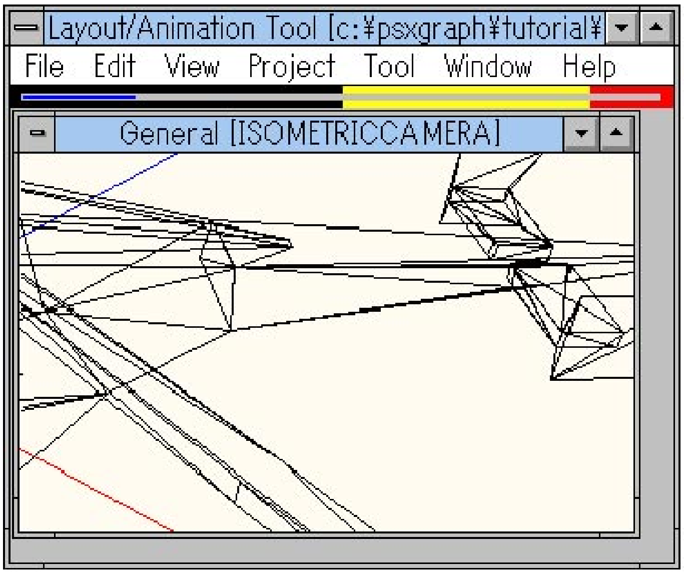


Figure 6-2-4 Swinging the Camera Up and Down, and to Right and Left

Holding down the Ctrl key, strike the right cursor key. The direction changes only a little. To fine-adjust the direction of the camera, strike the right or left cursor key while holding down the Ctrl key. Each time you strike the cursor key with the Ctrl key held down, the camera rotates by one degree.

Tilting the camera

You can tilt the camera as you tilt your head when you see a tilted object. Put the camera away from the helicopter to such a point where you can see the helicopter easily. To tilt the camera to the left, strike the left cursor key while holding down the Shift key. The helicopter is now tilted. To tilt it to the right, strike the right cursor key while holding down the Shift key. Each time you strike the right or left cursor key, the camera is tilted by ten degrees.

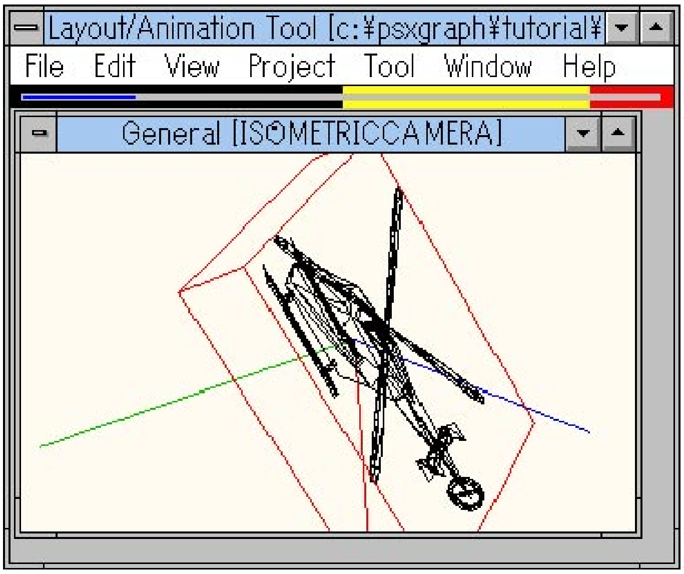


Figure 6-2-5 Tilting the Camera

Now, you can perform fine adjustment. While holding down the Shift and Ctrl keys, strike the right or left cursor key. In this case, the camera is tilted by one degree.

The simplest method of handling the camera has been described in this chapter. You do not have to memorize the Shift key and Ctrl key combination, etc. Practice repeatedly until you learn them by heart.

3 Import of DXF File

This section describes how to load a DXF file, and convert it into an RSD data format file using this tool. You do not have to read this section, and you may proceed to the next chapter if:

You do not have a DXF file.

You do not have to use a DXF file.

You use a CG software program capable of handling RSD data format.

You know how to use DSF2RSD.EXE.

You know how to use DXF2RSDW.EXE.

Animation Tool performs conversion by calling up DXF2RSD.EXE. This is a smiplyfied method, and it does not support all the functions of DXF2RSD.EXE. It is more convenient to use DOS for batch processing when converting a plural number of files. If you want to call up DXF2RSD.EXE from this tool, you must install the DXF2RSD.EXE file to the subdirectory ¥RSXGRAPH¥BIN in drive C beforehand. You cannot call this file from any other directory.

Before starting operation, cancel DOS prompt, DOSEXEC, etc. If you have ionized DOSEXEC, you cannot start DXF2RSD.EXE.

In this chapter, you do not use the helicopter. If it is displayed on the screen, close HRC by selecting the “Close” command on the file menu. Get a floppy disk containing a DXF file.

Opening a project

Open C:¥PSXGRAPH¥TUTORIAL¥ANIM¥TRY1. To determine the destination address for saving of data after conversion, you must open the project before import. If you found the helicopter displayed on the screen, you do not have to do so because the project is already open.

Importing

Select “Import” command on file menu. The following dialogue box appears:

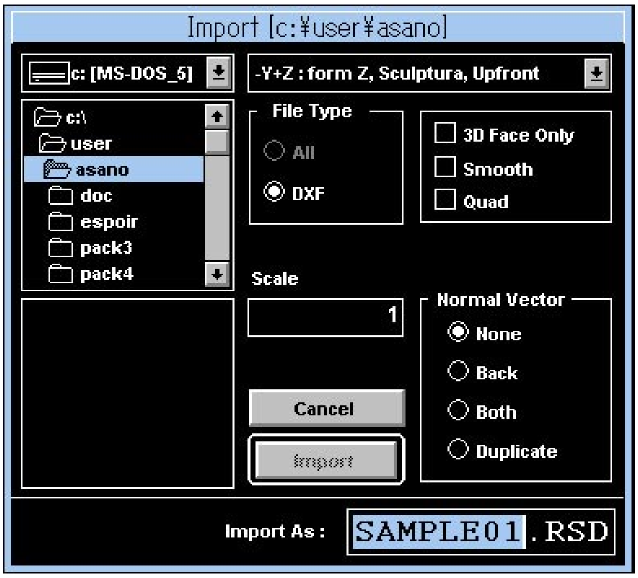


Figure 6-3-1 Import

Insert the floppy disk containing a DXF file to the floppy disk drive (e.g., drive A). Select drive A from the drive list. If the DXF filename (e.g., BALL.DXF) is displayed in the file field, select it. If not, move the directory. “BALL.DXF” in located in the root directory in drive A. Designate the filename after conversion as BALL.RSD, and click the “Import” button. The dialogue box disappears, and block DOS screen indicating the progress of conversion, etc. appears. When conversion is completed, the dialogue box appears again. This is instantly done if the amount of data is small.

Data after conversion is saved to the RSD directory of the currently open project.

This finishes import of the DXF file. This operation does not display data after conversion. If you want to display it, you have to add and save an object as described in the next section.

When saving a DXF file, you have to be careful about the position and posture of a model with respect to the modeler coordinate systems object coordinate system, local coordinate system, body coordinate system, etc.). As for the position, the result of import may extend beyond the General window, or displayed off the center. This causes no problem because you can move it using the function of parallel translation. However, if the posture is improper, you cannot correct it using the tool. For instance, if the front, rear, right, left, top, and bottom of the helicopter are tilted with respect to the modeler coordinate system, the tilted helicopter will be displayed inside the bounding box as a result of import. It is very difficult to adjust the posture of the helicopter in this status. Before saving the DXF file, be sure that the front, rear, right, left, top and bottom are parallel with the modeler coordinate system.

Adding and saving an object

The helicopter was displayed when you opened the HRC file in Subsection 6.1.3. However, this does not mean that helicopter shape data are saved in the HRC file. The role of the HRC file is to describe the relations with other objects, such as the objection position, posture, and hierarchical structure. Therefore, nothing would be displayed without the HRC file because it cannot be determined where and in what posture the object exists. This section describes how to add and save an object to the HRC file using “ball.rsd” created in the previous section.

1) Opening a project

Open C:¥PSXGRAPH¥TUTORIAL¥ANIM¥TRY1.

You do not have to call up this file if you did not discontinue work described in the previous section.

2) Create a working space

Select :New: command on file menu. General window appears. The object is not displayed yet.

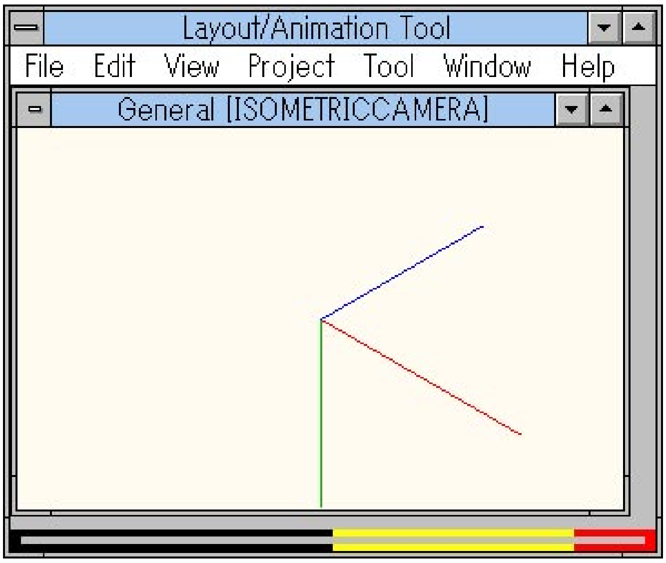


Figure 6-3-2 Creating a Working Space

3) Adding an object

Select “Add Object” command on file menu. The following dialogue box appears:

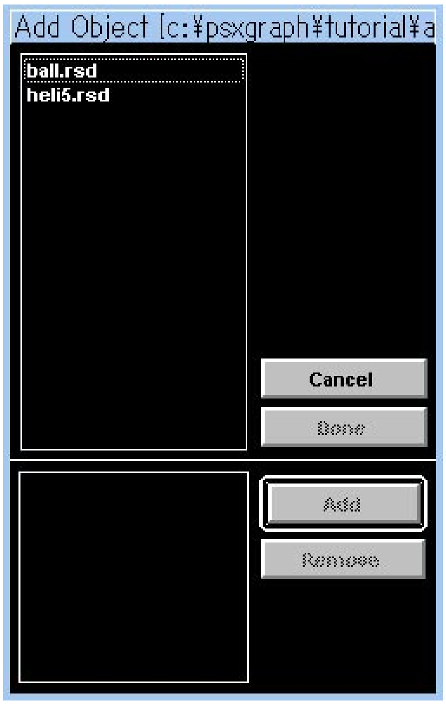


Figure 6-3-3 Adding an Object (1)

Select “ball.rsd” imported earlier, and click the “Add” button. The “ball.rsd” is displayed in the lower field. Click the “Done” button. The following dialogue box appears:

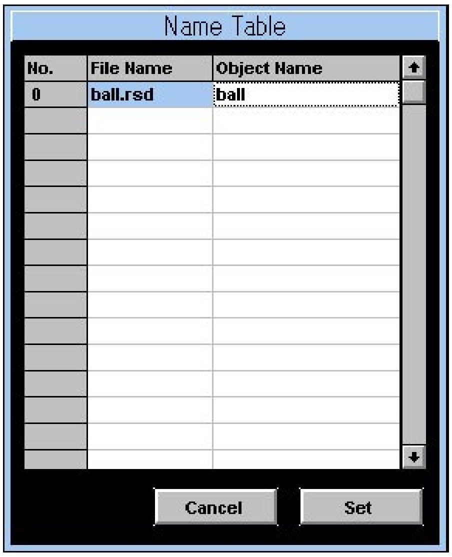


Figure 6-3-4 Adding an Object (2)

The RSD filename is displayed in the left field on the dialogue box. The right field is used to specify the object name. On the screen, “ball” is displayed as object name. For the time being, leave the object name as it is. Click the “Set” button. A ball is displayed on General window.

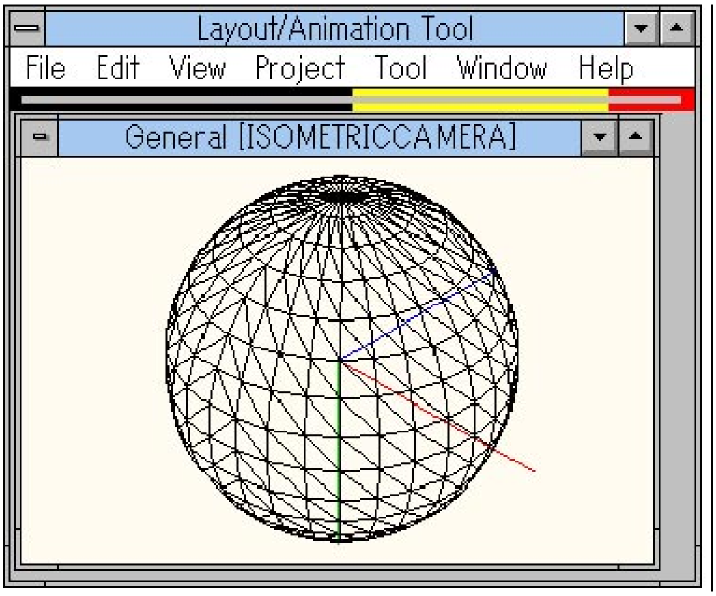


Figure 6-3-5 Added Object

4) Saving an object

Select “SaveAs” command on File menu. The following dialogue box appears:

Figure 6-3-6 Saving an Object

Enter the HRC filename to the box beside “Save As:”. Let us type “BALL.HRC”. If you click the Save button, a new HRC file saving the position and posture of the ball is created.

This completes adding and saving of an object. The HRC file thus created can be opened if you follow the same procedure you followed to display the helicopter for the first time. Of course, you can perform 3) and 4) for an open HRC file.

4 Creation of Hierarchical Structure

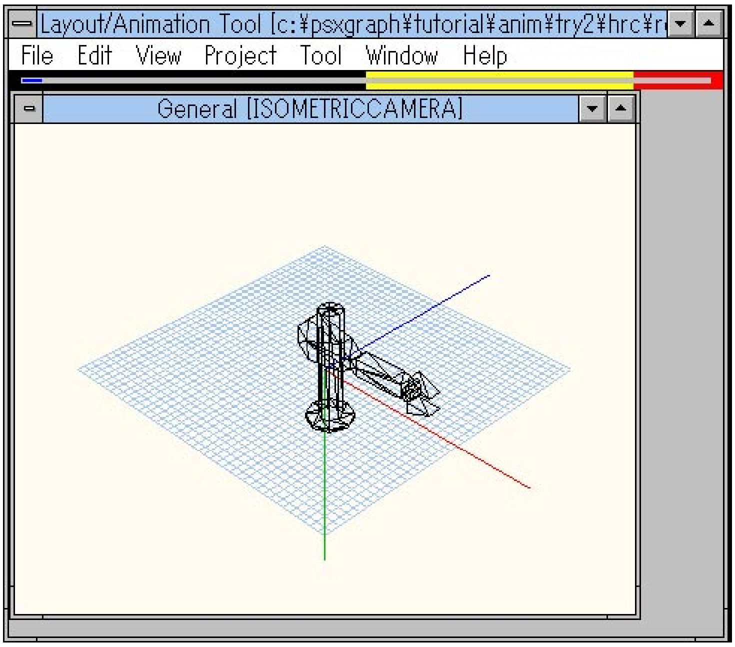


Figure 6-4-1 Creating a Hierarchical Structure

Open C:¥PSXGRAPH¥TUTORIAL¥ANIM¥TRY2, and then open ROBOTARM .HRC. A robot arm appears on the screen. The same image can be obtained if you assemble parts, such as the arm and mount, to get the same layout. However, it is too great a burden to do so each time you move the arm. To move the lower arm and hand according to bending of the elbow, you may use the hierarchical structure.

The root of a hierarchical structure is always the world coordinate system. Its child is for the mount, the child of the mount is for the shoulder, and the child of the shoulder is for the upper arm... The shoulder is, of course, a joint. In PlayStation, a joint is expressed in terms of parameters, such as transformation matrix. The parameters serve as the reference (coordinate system) in determining the position and posture of the object which is the child. In this too, the style of the object which can be treated as transformation matrix is called the “origin”. Normally, the origin is not displayed. To see it, you may select “Show Origin” on Vie menu. The origin is indicated by red, green, and blue orthogonal arrows. The red, green, and blue arrows stand for the X-, Y-, and Z-axes of the origin coordinate system, respectively.

The origin has a style of an object which can be handled as described above. However, it has no color or shape hen transferred to PlayStation. So, it uses hardly any memory space as compared with a model. The polygon indicator does not count the number of polygons in the origin.

In this sample, there is no origin where the elbow should be located. Let us apply an elbow, and complete the robot arm.

Creating an origin

Select CreateOrigin on Tool menu. The following dialogue box appears. Type “elbow” to name the object as “elbow”. Then, click the OK button.

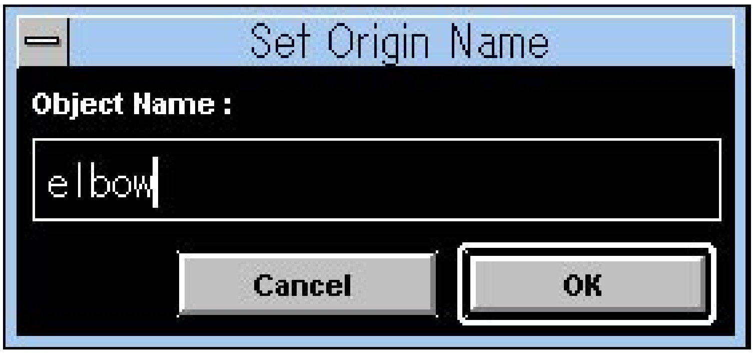


Figure 6-4-2 Creating an Origin (1)

The origin is displayed at the zero point of the world coordinate system. It is the elbow. Let us move the elbow to its position. Select the Front command on Window menu to call up Front window.

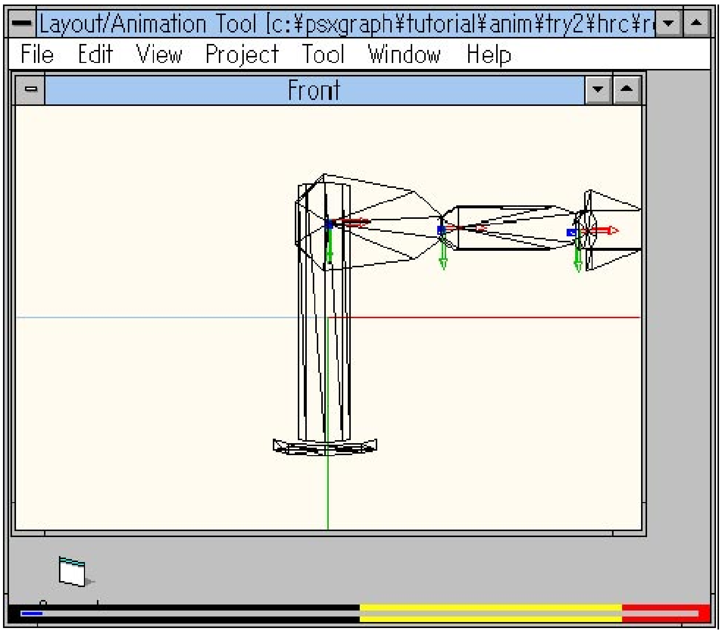


Figure 6-4-3 Creating an Origin (2)

Before selecting “elbow”, select the Origin icon on the tool palette, to restrict the object of manipulation to the origin. (While the origin icon is selected, there is no fear for selecting a model or camera by mistake.) Then, select the parallel translation icon to move the origin to the position of the arm without rotating it.

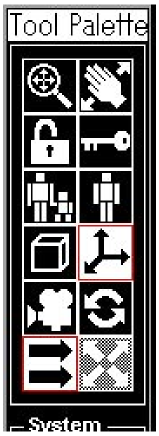


Figure 6-4-4 Restricting Object of Manipulation

During parallel translation on Front window, the default front camera coordinate system is in parallel with the world coordinate system. Therefore, if the object is moved up and down and to right and left on the Front window, the object moves on a plane which is parallel with the XY plane of the world coordinate system containing the center of the bounding box for the object. Because the center of the bounding box for the elbow is put on the XY plane of the world coordinate system, the elbow does not move apart from the XY plane of the world coordinate system as long as it is dragged up and dow and to right and left. Each axis of the elbow is made parallel to the axes of the correspondent world coordinate system. Especially, when you create other joints, be sure that their directions are the same. If their directions differ, trouble will be involved when they are manipulated. Furthermore, additional time will be needed when driving the joint structure using the program.

Linking

In this status, the elbow is put to its position, but the elbow and the lower portions will be left unmoved if the upper arm is lifted by turning the shoulder. So, let us employ the hierarchical structure. Select the Link icon indicated by the parent and child mark on Tool Palette. If “elbow” has not been selected, select it. To do so, select the origin icon indicated by a coordinate system mark, and then select “elbow” so that you may not select the upper arm or forearm by mistake. (At this time, the first clicking makes the window active, and the second clicking selects the object. This procedure is always followed whenever you return from Tool Palette.) The select object (elbow) is the child.

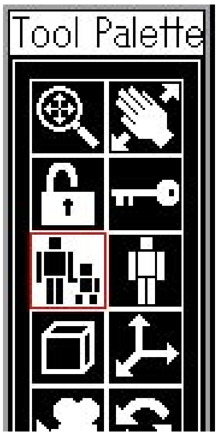


Figure 6-4-5 Linking

Then, let us select the parent. The parent of the elbow is the upper arm. Select the model icon indicated by a cube, and then select the upper arm. As long as the model icon is selected, there is no fear for selecting the camera or origin by mistake. Linking is successful if the upper arm is instantaneously wrapped with the green bounding box. When selecting the parent object, you may slowly press the mouse button and release it to watch it. You will not see the green bounding box even if you select a parent object.

Finally, let us link the forearm and elbow. In this case, the forearm is the child. While the selected object (elbow) indicated by the red bounding box exists, parent object selection mode is selected. Click the portion where no object exists to cancel this mode. Then, select the forearm. The forearm is the child. Then, select the elbow. When they are linked, click the Link icon to turn it off. The forearm and the subsequent objects are linked beforehand.

This completes the robot arm. Save the data to the HRC file immediately. Make it a rule save data to a file whenever you complete an important job. This tool does not have the function to warn you even if you fail to save data to an HRC file.

Check of linking

To check the hierarchial structure, select Object Browser on Tool menu. You should select “elbow” beforehand. The following dialogue box appears:

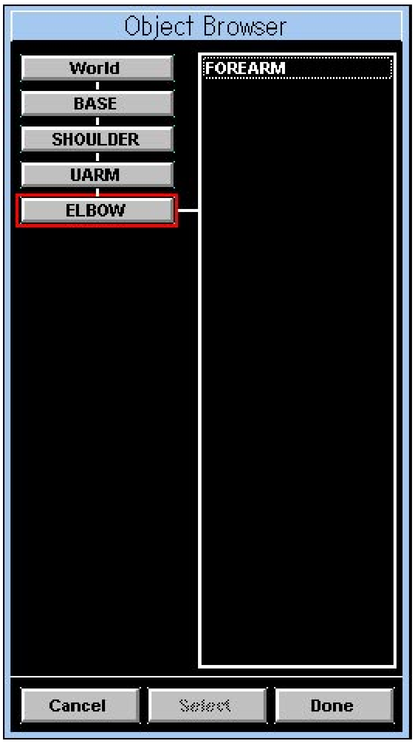


Figure 6-4-6 Check of Linking

On the left, the World, BASE, SHOULDER, UARM, and ELBOW buttons are located from the top to the bottom in that order. The red frame of the ELBOW button indicates that it is currently selected. They constitute the hierarchy of the objects above ELBOW. The left field indicates FOREARM, which is the child of ELBOW.

If you click any of the buttons on the left, the object having that name is selected. You may click them to circulate the hierarchy. If you select a name indicated on the right, and press the Select button, or if you double-click the name, the object having that name is selected. You can use this function to descend the hierarchy. This is a way of selecting an object. If numerous objects are used, it will be difficult to select a target object without touching other objects. In such a case, this browser function will be helpful. Now, click the Cancel button, to close ObjectBrowser.

Cancellation of linking

Let us see the procedure for cancellation of linking so that you can follow it when you make a mistake. The “Unlink” icon is indicated by a highlighted mark shown in the figure below. Select that icon.

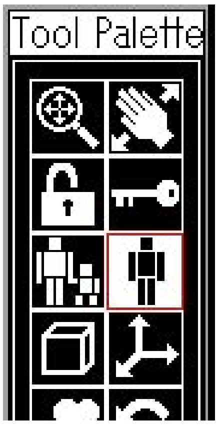


Figure 6-4-7 Cancellation of Linking

To cancel linking, select an object, and click it again. The selected object is separated from the parent. At this time, you do not have to select the parent object. The objects (child) below the selected object in the hierarchy are maintained as they are. Now, let us separate the elbow from the upper arm. Select the elbow if it has not been selected. Cock the elbow again after selecting it. It is instantaneously covered by the bounding box. The forearm, wrist, and hand which are located below the elbow remain the child and grandchild, etc. The world coordinate system serves as the parent for the separated object as was the case when the object was added and saved for the first time. You cannot break these relations. Call up ObjectBrowser, and click the world button. If linking is successfully cancelled, “elbow” should be displayed in the right field.

Moving the robot arm

Close the file once. Select "Close" on File menu. The, open ROBOARM.HRC. This is a finished robot arm. Let us bend the elbow. Select Origin on Tool Palette so that you may not select the upper arm or forearm. The elbow has only one rotation axis. If an object is to be rotated around only one axis, it is convenient to inhibit rotation around other axes. Because the Z-axis is used as the rotation axis for the elbow, select "Z Only" on Tool Palette. If the rotation axis is definite, you do not have to set the starting point for dragging to a ridgeline which is parallel with the axis of the bounding box when you are going to start rotation. The starting point for dragging may be any place on the bounding box. Therefore, you can rotate the elbow on any window. It may seems the same with the helicopter even when the elbow is rotated. However, if the mouse button is released, the elbow rotates and, the all the objects below the elbow in the hierarchy moves accordingly. Try to move other joints. You will now understand why the directions of the origins were made the same. If the directions of the origins for the joints are different, you will have to manipulate Tool Palette.

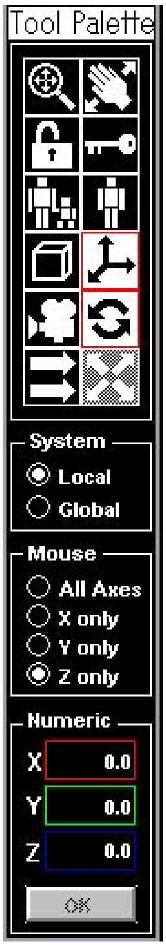
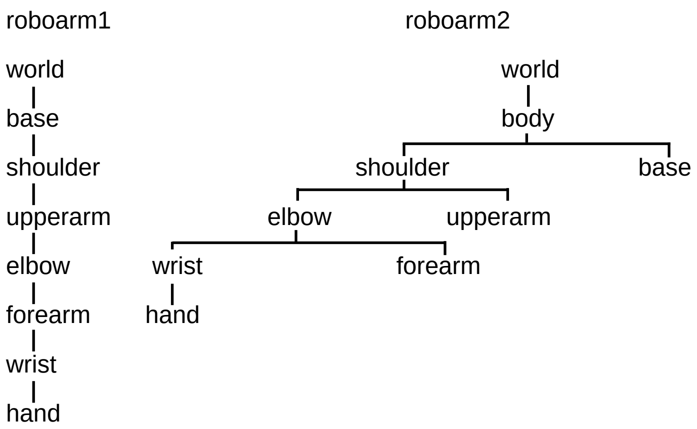


Figure 6-4-8 Moving the robot arm

Another method

Let us try to extend the mount upward. On General window, drag the mouse using the top of the pillar as starting point. When the mount has been extended, rotate the shoulder on Front window. You will see that the shape of the bounding box is distorted. The degree of distortion is proportionate to how long the mount was extended. Release the mouse button when it is put to a portion where distortion is extreme. If a model having a child is enlarged or shrunk in this manner, the child objects are adversely affected. This is attributable to the library specification of PlayStation. You cannot avoid it. Close ROBOARM1.HRC, and open ROBOARM2.HRC. Now, the child and grandchild are not affected even if you rotate the shoulder. The differences between the hierarchical structures of ROBOARM1.HRC and ROBOARM2.HRC are shown below:



If a hierarchical structure contains a model which is likely to be enlarged and shrunk, it is essential not to create a child or grandchild below that model in the hierarchical structure.

5 Creation of Animation (Sequences)

This section describes how to create an animation.

Let us see what meanings the term “animation” and “sequence” have. In a motion picture, a picture of an actor’s back view and a zoomed picture of his face as it is turned are taken separately, and connected later. The tool provides the concept of sequences so that an animation can be made up of some portions. That is, sequences constitute an animation, and a series of sequence is called an animation. It is not difficult to understand. A part of animation can be an animation itself. You may think that “Sequence = Animation”.

Let us see some examples.

First, open the project C:¥PSXGRAPH¥TUTORIAL¥ANIM¥TRY3, and open TE.HRC.

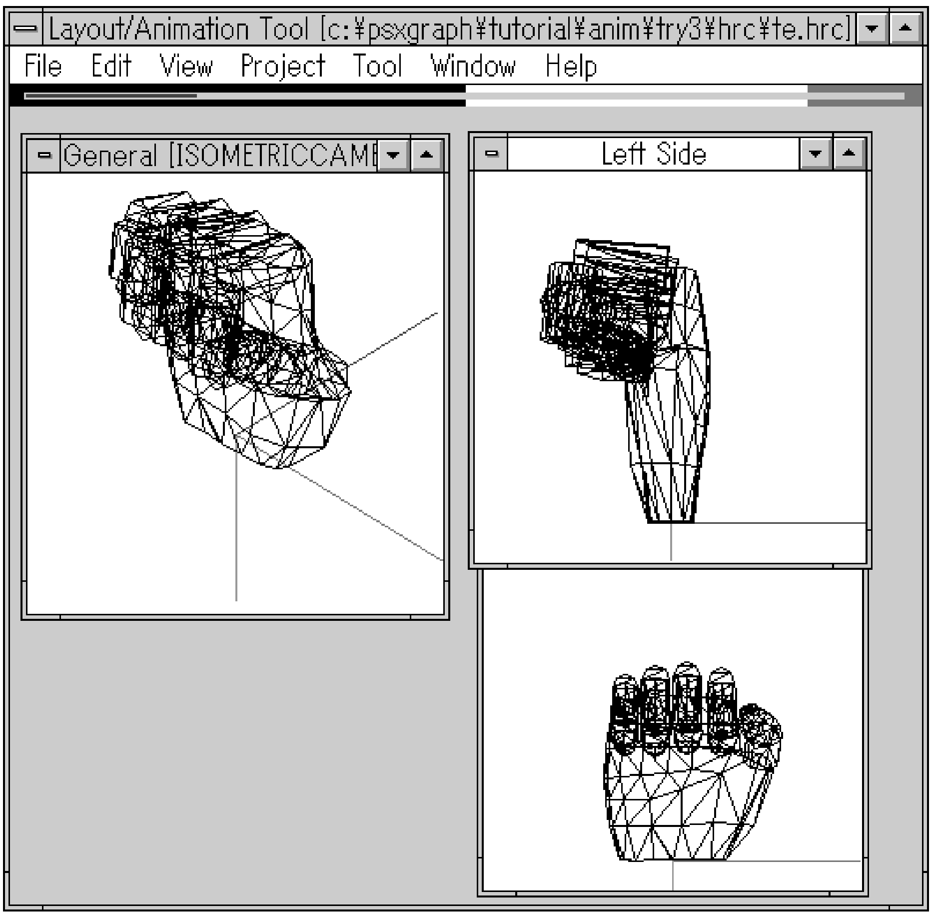


Figure 6-5-1 Sample Data “Hand”

The hand shown on the screen has the following hierarchical structure. (It is shown in General, Left Side, and Front windows in Figure 6-5-1.) As in the example of the robot arm, the joint of each finger is expressed as origins so that the movement may be transmitted from the root of each finger to the tip.

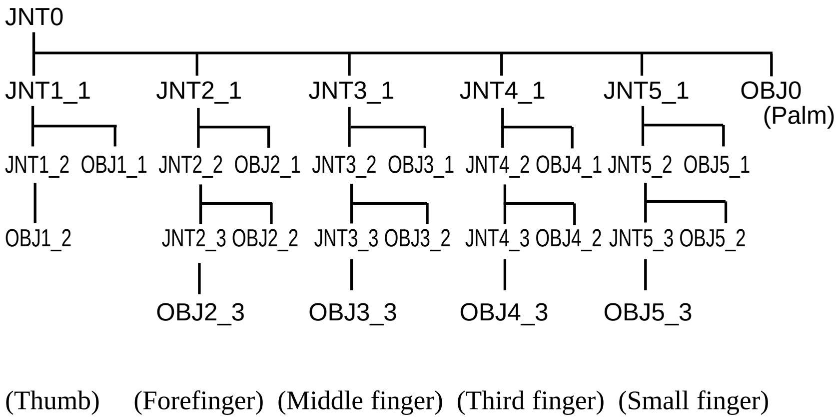


Figure 6-5-2 Hierarchical Structure of Sample Data “Hand” 1

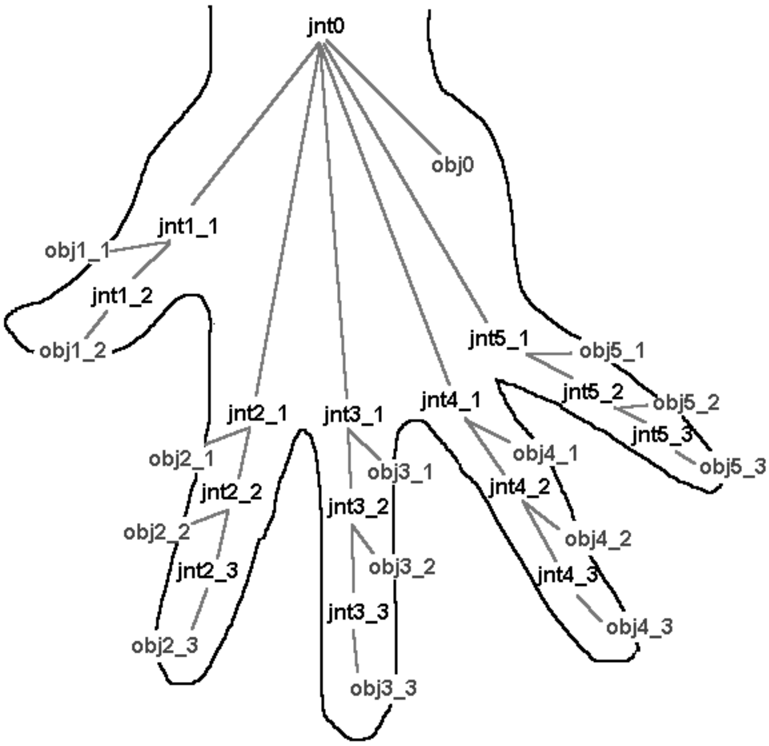


Figure 6-5-3 Hierarchical Structure of Sample Data “Hand” 2

Using the data on the hand, let us create an animation of “janken”.

Making preparations to create sequences

Let us create sequences first. It is advisable to use names representative of the animation as sequence names so that they can be easily identified.

The project C:¥PSXGRAPH¥TUTORIAL¥ANIM¥TRY3 contains some sequences.

- N-P-N: From lowering of the hand to show “paper”, to raising of the hand
- N-C-N: From lowering of the hand to show “scissors”, to raising of the hand
- N-G-N: From lowering of the hand to show “stone”, to raising of the hand
- APPEAR: The hand walks, and appears.
- DISAPP: The hand jumps to retire.

In addition to these, let us create a sequence in which the hand is swung twice with a cry “jaankeen”.

Select the New Sequence command on Project menu. Type “JAANKEEN” in place of “AMPLE01” on the dialogue box, and click the Create button. The dialogue box disappears, and KeyframeRecorder appears.

Now, preparations for creating a sequence named “JAANKEEN” have been made.

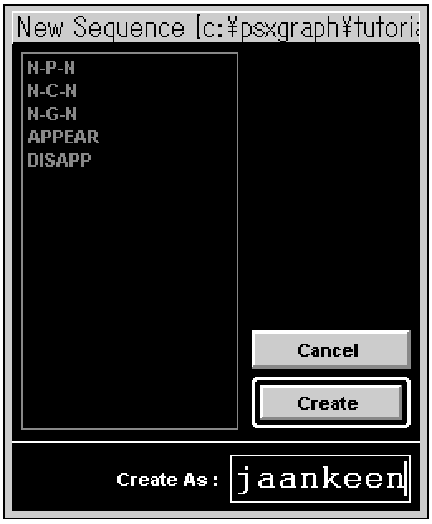


Figure 6-5-4 New Sequence Dialogue Box

Creating a keyframe

Before creating an animation, you have to create keyframes. An animation is created by interpolating the created keyframes.

When Keyframe Recorder is displayed on the screen, the slider is put to 0.

Figure 6-5-5 Keyframe Recorder Immediately after Calling

To this position, create the pose of lowering of the hand as keyframe. Currently, the hand shows “stone”. Rotate the origin named “JNT0” located at the highest level in the hierarchical structure at the position of the wrist by 30 degrees. You may use the function of rotation by numerical values. This is the fundamental position.

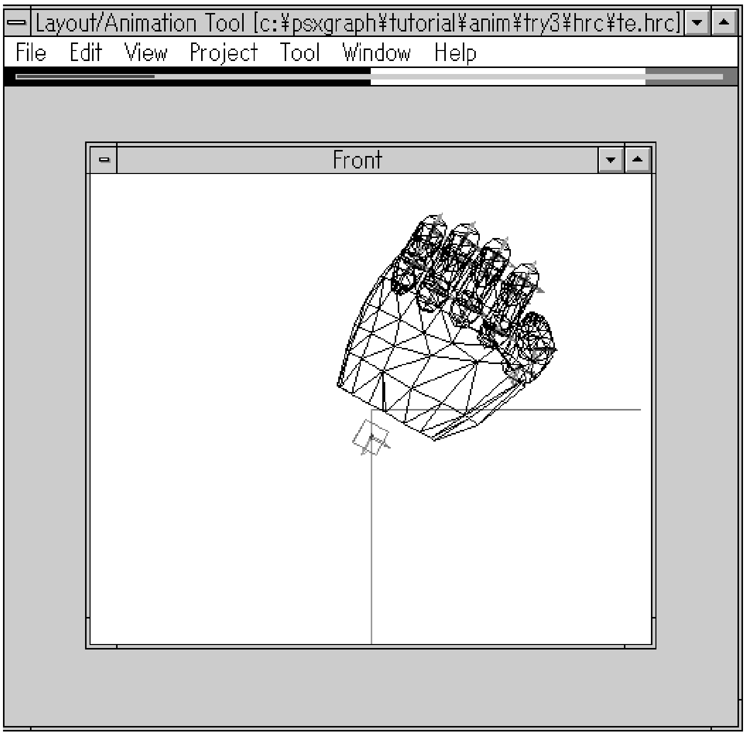


Figure 6-5-6 Fundamental Position

Note that this does not complete preparation of the keyframe. To create a keyframe, it must be saved. If you should fail to do so, the created pose would not be saved.

To save the keyframe, turn on the “Propagate” button. (It is turned on when Keyframe Recorder is called up.) Be sure that the “Create” button is green, press the Record button (marked with a red circle). Do not forget to turn on the “Propagate” button in the above procedure. If the “Create” button is not green, click it once. Press the Record button, and the green keyframe marker is indicated at the “0” position.

Figure 6-5-7 Keyframe Recorder Immediately after Saving of First Keyframe

Now, this keyframe is finished.

Let us create a keyframe for lowering the hand.

Move the slide on Keyframe Recorder to the “20” position. Note that you have to move the slider to the position of the keyframe which you are going to create before starting work. If you move the slider without saving the created keyframe, all the work will end up in a failure.

Let us create a pose of lowering the hand by rotating JNT0 by -90 degrees around the Z-axis.

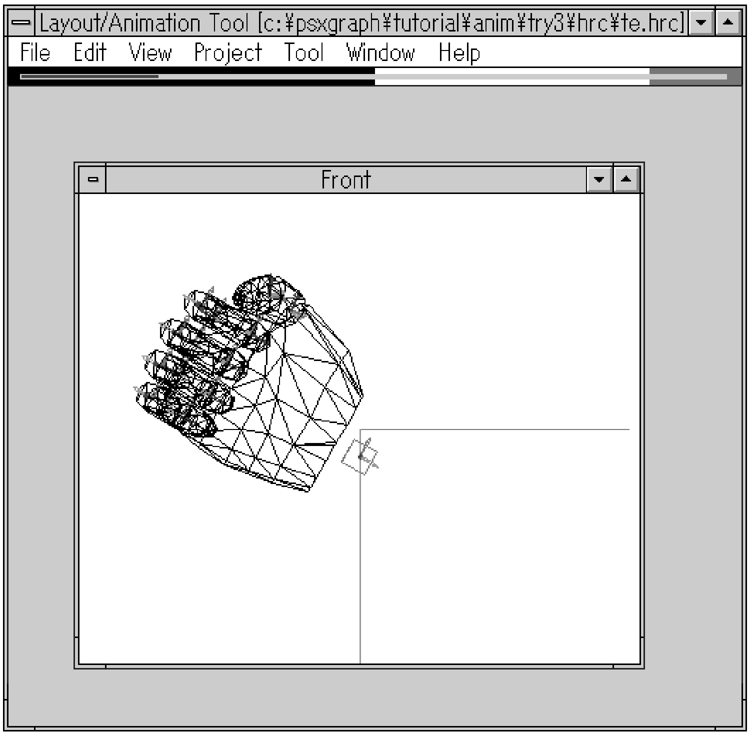


Figure 6-5-8 Position of Lowering Hand

Let us save the keyframe again. Now, be sure that neither the "Create" button, nor the "Kill" button on the screen is selected. Currently, the "Create" button may be displayed in green. Then, you should click the "Create" button so that both of these buttons may be indicated in yellow. You should not mind if the "Propagate" button is on. However, put turn it off so that unnecessary keyframes may not be created. As you did earlier, press the Record button. Now, the yellow keyframe marker is indicated at the "20" position. Thus, another keyframe has been saved.

Figure 6-5-9 Keyframe Recorder Immediately after Saving of Second Keyframe

In the similar manner, move the slider on Keyframe Recorder to the "40" position, and rotate the hand around the Z-axis by 60 degrees from the original position to get the same pose which existed when TE.HRC was opened. This is the starting position for lowering of the hand for the second time.

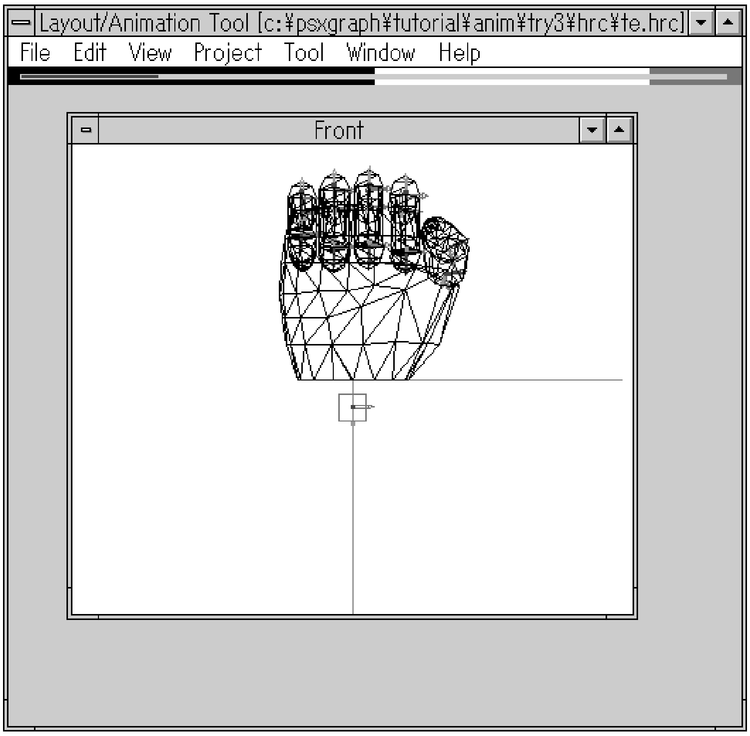


Figure 6-5-10 Starting Position for Lowering Hand Second Time

Figure 6-5-11 Keyframe Recorder just after the third keyframe registration

At the “60” position, let us create a pose for end of lowering of the hand for the second time. This pose is the same as that for the end of lowering of the hand for the first time. So, all that you have to do is to copy the keyframe at the “20” position to the “60” position. Click the keyframe marker at the “20” position, and move the slider to the “20” position. While holding down the Ctrl key, drag the slider to the “60” position, and release it.

Figure 6-5-12 Keyframe Recorder with Slider Moved to “Copy Destination” Position

Now, press the Record button. The keyframe has been copied.

Figure 6-5-13 Keyframe Recorder Immediately after Saving of Fourth Keyframe

In the similar manner, copy the keyframe at the “0” position to the “80” position.

Figure 6-5-14 Keyframe Recorder Immediately after Saving of Fifth Keyframe

Thus, we have created five keyframes. Return the slider to the “0” position, and press the “>” key to play the animation. Have you successfully created an animation which swings the hand to play “janken”?

Let us see differences among the green, yellow, and red keyframe markers. (You have not seen the red keyframe marker. It is displayed if you pres the Record button when the “Kill” button is displayed in red.)

The green keyframe marker indicates that the object or origin appears (is created) in that frame. The red keyframe maker indicates that the object or origin appears (is created) in that frame. The yellow frame marker indicates that the object or origin appeared (was created) in any previous frame. In most cases, therefore, a sequence is so crated that the marker for the first keyframe may be green, and the marker for the subsequent keyframes may be yellow. Unless you create an object or origin, it will not appear on the screen even if you try to display that data on a program displayed on PlayStation. Remember that you turned on the “Propagate” button when you created the keyframe at the “0” position. You did so to create not only JNT0, but objects or origins below JNT0 in the hierarchical structure.

What you should bear in mind is that you must create a parent for data having a parent-child structure before or at the same time when you create a child. Otherwise, an error will occur when you try to load TOD data created from that sequence into a display program for PlayStation to display it on the screen.

Saving to a TOD file

Now, let us save the created animation to a TOD file. It is advisable to use a name representative of the animation as the TOD filename. Let us use JAANKEEN.TOD. (Be careful because there is a file having a similar name “JAANKEN.TOD.) Enter 0 into Start and 80 into End, and press the “Export” button.

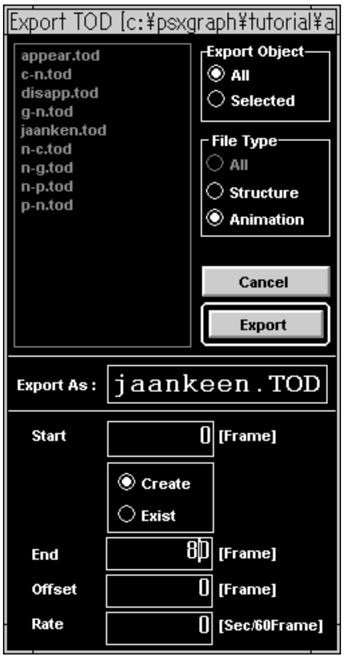


Figure 6-5-15 Export TOD Dialogue Box

Now, JAANKEEN.TOD containing animation data for total eighty-one frames has been created under the directory C:¥PSXGRAPH¥TUTORIAL¥ANIM¥TRY3¥TOD.

Viewing TOD data

To view the TOD data you created, you need a programmer tool. For this purpose, you can use TODVIEW, which is a sample program used to view TOD data. (It is saved as C:¥PSX¥SAMPLE¥GRAPHICS¥TOD¥TODVIEW.) Let us view the TOD data using TODVIEW.

Sample data provided together with TODVIEW is the same data as you created by following the procedure described above. Referring to the attached README, execute the program. On the monitor, you will see the hand showing “stone” swing twice with a cry “Jaankeen”.

Among the data under the directory, the following data are created by the 3D graphic tool, and transferred to the program:

- TOD data (“JAANKEN.TOD” in TODVIEW)
- TMD data (“TE.TMD” in TODVIEW)
- TMD ID list (“TE.H” in TODVIEW)
- TIM data (Not used in TODVIEW)

If you replace these files with ones you created by yourself, you can check the details of the data you created. Among these data, you already created TOD data in the above-mentioned practice. However, you must get 3D model data (TMD data/TMD ID list) used in the created animation. To create these data, select the TMD command on File menu. (For details, see the description of Export TMD in CHAPTER 4, “ANIMATION TOOL”, in PART III, “REFERENCE”. Now, the TMD data and TMD ID list are created.

CHAPTER 7

MIME ANIMATION

This chapter describes MIMe animation, which is a technique unique to PlayStation.

1 What is MIMe Animation

How the famous dinosaur in the demonstration program for PlayStation was created? Even if use the techniques described in CHAPTER 6, you cannot smoothly change the shape of the mouse, and move the feet without dislocating any joint. The fierce features and waling patterns are synthesized real-time. It is an animation technique unique to PlayStation called "MIMe animation". It has great power of expression, and is suitable for soft expressions. However, it imposes a great load on execution. The performance of PlayStation is essential to this animation technique.

Let us see the principle of MIMe animation.

Principle of MIMe animation

In MIMe animation, the coordinates of each vertex of a 3D model is separately moved real-time. (See NOTE.) Furthermore, its movement is controlled using parameters. So, you can simulate a wavering motion, and a fist delicately bouncing after hitting an object. This method greatly differs from the conventional method of animation in which the position (using a set of vertexes) of vertexes of an object is moved from one keyframe to another.

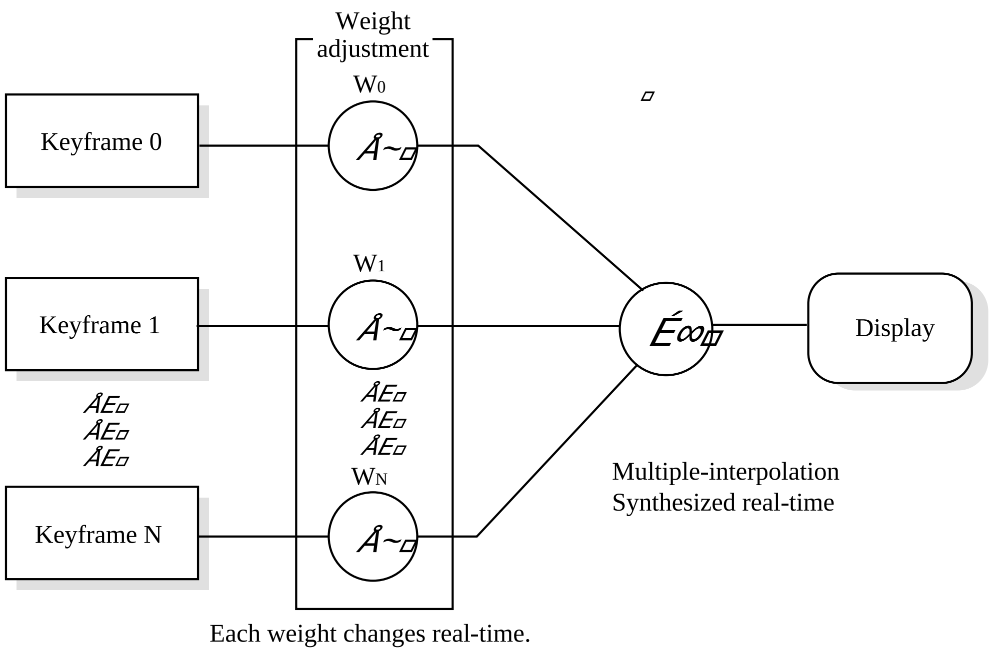


Figure 7-1-1 Principle of MIMe

NOTE: The concept of MIMe is not restricted to vertexes. It can be applied to variations, such as normal, angle, and UV coordinates of a texture. For simplicity, only vertexes are discussed in this section.

Keyframes are also used in MIMe animation. In MIMe, indefinite patterns are created by interpolating keyframes by applying weight factors to each keyframe, and obtaining weighted average (multiple-interpolation). “To obtain weighted average” or “to perform multiple-interpolation” means to create a average frame by blending two or more keyframes at various ratios. MIMe animation performs such interpolation real-time. (See Figure 7-1-1.)

To perform interpolation of two keyframes, it is convenient to have finite differences rather than hold two keyframes as they are. MIMe uses one original model, and data on finite differences from the original model, and performs interpolation using them.

Imagine weight factors arranged according to the time series. It will constitute a waveform, and expresses the change of ration at which particular keyframes are blended. This waveform data is called keyframe “control waveform”. It is an important parameter which characterizes movement.

For instance, if the weights of two keyframes change as shown in the left view in Figure 7-1-2, the effect will be the same as that of a linear keyframe animation. The right view shows spline-interpolated keyframe animation.

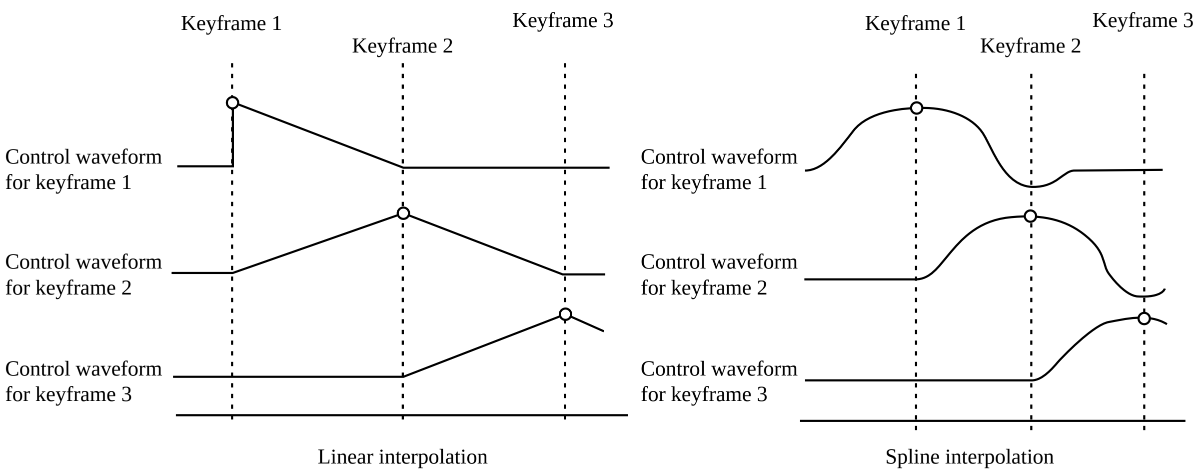


Figure 7-1-2 Control Waveforms

Suppose a certain keyframe has a control waveform shown in the figure below, a real effect of inertia can be expressed because the object returns after going a little too far.

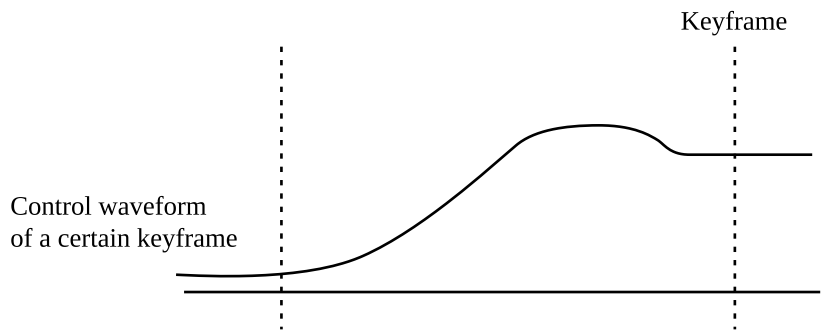


Figure 7-1-3 Control waveform with overshoot

Thus, MIMe expands the range of keyframe animation, and expresses more complicated movement.

Data necessary for MIMe

MIMe animation requires the following data:

○ 3D model (original model)

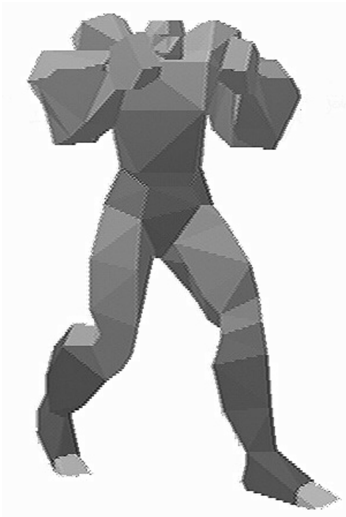
○ Keyframes (modified modes), and data on finite differences from the original model

○ Control waveform data (as many as the number of the modified models)

How many keyframes there may be, only one original model is stored in memory. Models in other keyframes can be restored using the original model and finite differences. In this manner, MIMe is advantageous in memory efficiency.

Creation of model data for MIMe animation begins with creation of some poses. Let us see the procedure for creation of MIMe data using sample data. The directory for the sample data is TOTORIAL\MIME.

2 Creation of Boxer



Let us create a 3D model "man" as the original model for MIMe, and let him fight as a boxer. You may use the sample model. The directory is TUTORIAL¥MIME¥RSD. Some DF files and RSD data are saved under this directory.

Figure 7-2-1 Boxer

The procedure for actual modelling is described below.

First, create the 3D model of the boxer as one object. That is, create a shape which is not cut off at the joints as shown in Figure 7-2-1, and save it as one DXF file.

1) Create a original mode, which is a fundamental pose, and save it as DXF. Name the file as BOXER0.DXF. At the same time, save the format of the modeler. The original model will be loaded many times because it is modified into various models.

2) Based on the original model, create "a showy pose" by changing the positions of the arms. Repeatedly move the vertexes and polygons, and save the result as BOXER.DXF. This is called the "modified model". It resembles a keyframe.

3) Create another modified model. It is advisable to load the original model again for modification. Modify the original model in a manner different from 2) above, and save it as BOXER2.DXF.

TIPS:

MIMe is suitable for soft movement. A real effect can be obtained if you change the swelling of the muscles according to the positions of the arms.

4) By repeating the procedure described in 3) above, create as many modified models as necessary. In this example, four models (BOXER1.DXF through MOXER4.DXF) are created.

5) Now, convert the original models and modified models into TMD.

```
> DXF2RSD -g BOXER?.DXF
```

Thus, five RSD data have been created. To express smooth surfaces, smooth shading (-g option) was performed. Give names to RSD data, and convert them into TMD.

```
> RSDLINK -o BOXER0.TMD BOXER 0

> RSDLINK -o BOXER1.TMD BOXER 1

> RSDLINK -o BOXER2.TMD BOXER 2

> RSDLINK -o BOXER3.TMD BOXER 3

> RSDLINK -o BOXER4.TMD BOXER 4
```

We paste no texture in this section. MIMe animation can be created by creating a model with textures using Material Data. At this time, you may paste textures to the original model. MIMe modifies the textures real-time.

Now, all the model data for MIMe have been prepared. Let us give finite differences. MIMEFILT.EXE is a tool for that purpose.

3 MIMEFILT.EXE

MIMEFILT.EXE outputs the finite differences for vertex data and normal data based on the original TMD file and modified TMD file(s). This program is used in the following manner:

```
> MIMEFILT -n BOXER0.TMD BOXER1.TMD BOXER2.TMD BOXER3.TMD BOXER4.TMD
```

As a result, the BOXER0.VDF and BOXER0.NDF files ar created. They are the “vertex difference file” and “normal difference file”. These files contain the array of differences from individual X-, Y-, and Z-coordinates (components). (The number of finite differences has been a little reduced.) For details, see Section 2.3, “MIMEFILT.EXE” in PART III, “REFERENCE”.

The n- option is used to specify normal MIMe. Normal MIMe synthesizes luster and shadows on the surfaces real-time.

Thus, finite difference data has been prepared.

4 Execution of MIMe Animation

In addition, MIMe animation requires control waveform data. It determines at what speed each modified model changes its shape, and at what ratio modified models are synthesized. Generally, control waveform data is defined as a waveform in the range between 0 and 1. Notice that 1 is expressed as 4096 because fixed point calculation is performed inside PlayStation. An example of a control waveform is given below. It is actually used to express blinking of eyes of the dinosaur.

```
static int blinktable[120] = {
           0, 1024, 2048, 3072, 3072, 3072, 3072, 2048, 2048, 2048,
        1024, 1024, 1024,  768,  768,  512,  256,  128,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0, 2048, 4096, 4096, 4096, 4096, 4096,
        3072, 2048, 1024,    0,    0,    0, 2048, 4096, 2048,    0,
           0,    0,    0,    0,  512, 4096, 3072, 2048, 1024,  512,
           0,    0,    0,  512, 4096, 3072, 2048, 1024,  512,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           0,    0,    0,  512, 1024, 2048, 2048, 2048, 2048, 2048,
        1024,  768,  512,  256,    0,    0,    0,    0,    0,     0
};
```

Figure 7-4-1 Control Waveform Array

To move the boxer, you need a program. You can utilize the sample program "MIME5" attached to "Program Tool". Ask a person responsible for the entire program for cooperation, and execute it on the PlayStation board. Make sure that the boxer lands a punch according to the controller input. If you press two or more buttons, a shape created by synthesis of some poses is obtained. This is the effect of multiple-interpolation of MIMe. In this manner, MIMe animation allows creation of indefinite patterns by blending keyframes at any desired ratio.

In the sample program MIME5, step functions are entered as controller input, and a control waveform is used for convolution, and the result of convolution is used as a weight factor. That is, the two waveforms are multiplied, and a control waveform for interpolation is generated real-time as shown in Figure 7-4-2.

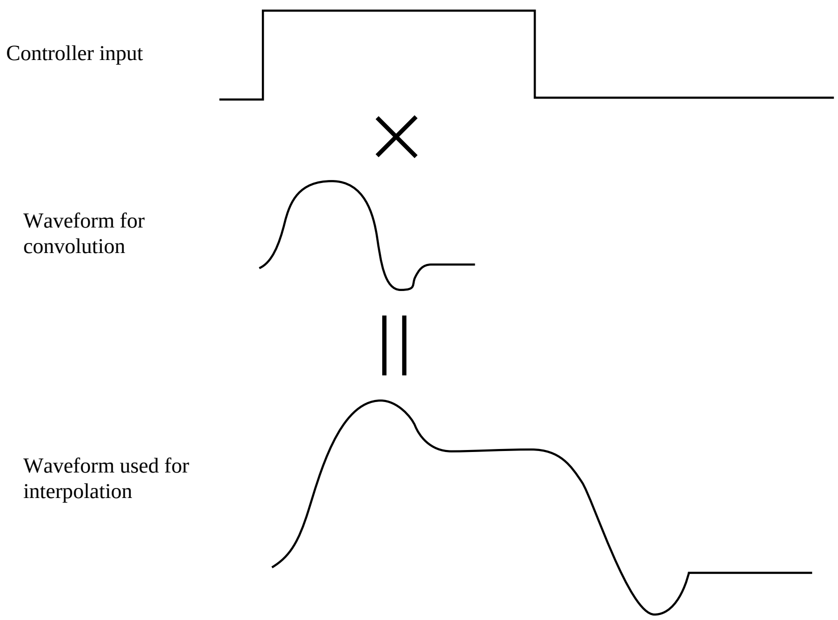


Figure 7-4-2 Control Waveform and Convolution

The program becomes simple if convolution is employed to respond to controller input. In the case of MIMe animation which does not resort to the controller, the control waveform is directly used as a weight factor.

5 More about Use of 3D Modeler

You must use care when using the 3D modeler to create MIMe model data. To change the shape of a 3D model, you have to move the vertexes, and you must not increase or decrease the number of vertexes. Furthermore, you must not change the order of vertexes because correspondence of vertexes can only be known from their order. This may seem to be a strict restriction. However, no problem will arise as long as you simply move vertexes and planes (sets of vertexes).

Fundamentally, MIMe performs linear interpolation. If you directly apply rotary motion as described in the example of “janken” in CHAPTER 6, a problem, in which figures will become too short during execution, will arise.

In such a case, you will have to take the following actions:

1. To insert a keyframe for interpolation, and

2. To perform curve-interpolation (spline, etc.) of weight factors

In addition to MIMEFILT.EXE, 3D Graphic Tool includes MIMESORT.EXE, which minimizes the size of differential data. It reduces the size of difference data, and assures efficient MIMe animation. For details, see Section 2.4, “MIMESORT.EXE” in PART III.

PART 3

REFERENCE

CHAPTER 1

2D UTILITY

1 BMP2TIM.EXE

1. General

This software converts the Windows BMP file into the PlayStation image data file (TIM).

2. How to use

BMP2TIM [options] BMP-files ...

Input of the Windows3.0 BMP format image data file creates the PlayStation TIM file. The wildcard is available for the argument. You can input and convert multiple files at a time. Omission of the ".BMP" of the filename is allowed.

3. Options

-o output-file

Specify an output filename. The default creates a filename identical with the input filename at the current directory, except extension which is converted into the ".TIM".

-org x y

Specify the VRAM image data area origin. The default is (0,0).

-plt x y

Specify the VRAM pallet (CLUT) area position. The default is (0,480).

-t

Turns the transparent control bit ON against any other color than (0 0 0). Any other color than (0 0 0) turns semitransparent. The default lets all colors turn nontransparent.

-b

Turns the transparent control bit OFF against (0 0 0).(0 0 0) turns transparent. The default lets (0 0 0) turn black.

-v

Shows detailed information about conversion.

4. Constraints

- The BMP compressed format is unavailable for the current version

2 PICT2TIM.EXE

1. General

This software converts the Macintosh PICT file into the PlayStation image data file (TIM).

2. How to use

PICT2TIM [options] PICT-files ...

Input of the Macintosh PICT format image data file creates the PlayStation TIM file. The wildcard is available for the argument. You can input and convert multiple files at a time.

3. Options

-o output-file

Specify an output filename. The default creates a filename identical with the input filename at the current directory, except extension which is converted into the ".TIM".

-org x y

Specify the VRAM image data area origin. The default is (0,0).

-plt x y

Specify the VRAM pallet (CLUT) area position. The default is (0,480).

-t

Turns the transparent control bit ON against any other color than (0 0 0). Any other color than (0 0 0) turns semitransparent. The default lets all colors turn nontransparent.

-b

Turns the transparent control bit OFF against (0 0 0).(0 0 0) turns transparent. The default lets (0 0 0) turn black.

-v

Shows detailed information about conversion.

4. Constraints

The current version has the following constraints.

- Convertible PICT data is limited to the Pixmap.
- 32-bit PICT data cannot be converted.
- JPEG compressed PICT data cannot be converted.

3 RGB2TIM.EXE

1. General

This software converts the RGB format image data file into the PlayStation image data file (TIM).

2. How to use

RGB2TIM-size w h [options] input-file

Input of the RGB format image data file creates the PlayStation TIM file. For -size, you must specify numbers of pixels for the width and height of the input image. The input data is the 24-bit interleaved RGB data. The wildcard is available for the argument but you can convert only one file at a time.

3. Options

-o output-file

Specify an output filename. The default creates a filename identical with the input filename at the current directory, except extension which is converted into the ".TIM".

-org x y

Specify the VRAM image data area origin. The default is (0,0).

-plt x y

Specify the VRAM pallet (CLUT) area position. The default is (0,480).

-mode 4 | 8 | 16 | 24

Specify a pixel size for the output TIM data. The default is the 16-bit mode.

-t

Turns the transparent control bit ON against any other color than (0 0 0). Any other color than (0 0 0) turns semitransparent. The default lets all colors turn nontransparent.

-b

Turns the transparent control bit OFF against (0 0 0).(0 0 0) turns transparent. The default lets (0 0 0) turn black.

-skip n

Specify the number of bytes which are skipped from the head of file. The default is 0.

-v

Shows detailed information about conversion.

4. Supplement

- Too many colors of the RGB data which is inputted may cause conversion of it to 4 or 8-bit mode to be impossible. In such case, reduce beforehand the number of colors with Photoshop and others.

4 TIMEXP.8BE

1. General

Features

TIMEXP.8BE is a plug-in module which allows you to output image which is created and amended with Adobe Photoshop to DTL-H201A and view it on a video monitor. With this, you can check how the image data is displayed on the PlayStation system.

This software has the following features.

- **Supports 4 color modes**

  The software supports all image data modes (4/8-bit CLUT and 16/24-bit Direct) that the PlayStation system can deal with.

- **Supports 8 display modes**

  The software supports all display modes that the PlayStation system can deal with. Change of the VRAM display start address allows you to display any VRAM area.

- **Allows you to specify the VRAM location**

  The software allows you to load the Display area, Texture page and Texture CLUT to any VRAM address. With this, you can also turn on/off Texture mapping to the Display area.

- **Supports the transparent control bit (partly)**

  The software provides you with a function to use the transparent control bit (STP) unique to the PlayStation system.

Operating requirement

TIMEXP.8BE is subject to the Adobe Photoshop operating requirement because it is a module which is started up from Adobe Photoshop. General matters of Adobe Photoshop are not explained here because this manual is written for those who are familiar to its operating procedure. For details of Adobe Photoshop, refer to instruction manuals and others attached to Adobe Photoshop.

System configuration

In order to use TIMEXP.8BE, the following software is required.

- Adobe Photoshop 2.5J or more

2. Preparation

Installation of program

Copy TIMEXP.8BE at C:\PSXGRAPH\BIN to PLUGINS subdirectory in a directory in which PHOTOSHP.EXE was installed.

Setup of operating environment

Start up ABOARD.EXE, a tool for setting the DTL-H201A board address and set the DTL-H201A address.

Check of installation and setup

Starting up Adobe Photoshop and selecting “HELP/ABOUT PLUG-IN/TIM DISPLAY...” option from the Menu bar to open the About box allows you to check the address which is set. (The default address is 0x1340.)

3. Operations

In order to view the Adobe Photoshop image on DTL-H201A

Select a window you want to access and then output the Adobe Photoshop image information to DTL-H201A with "TIM DISPLAY..." command to view it on the display.

■ How to use "TIM DISPLAY..." command

1. Select "OUTPUT PLUG/TIM DISPLAY..." option from the file menu to access the TIM Export Options dialog box.

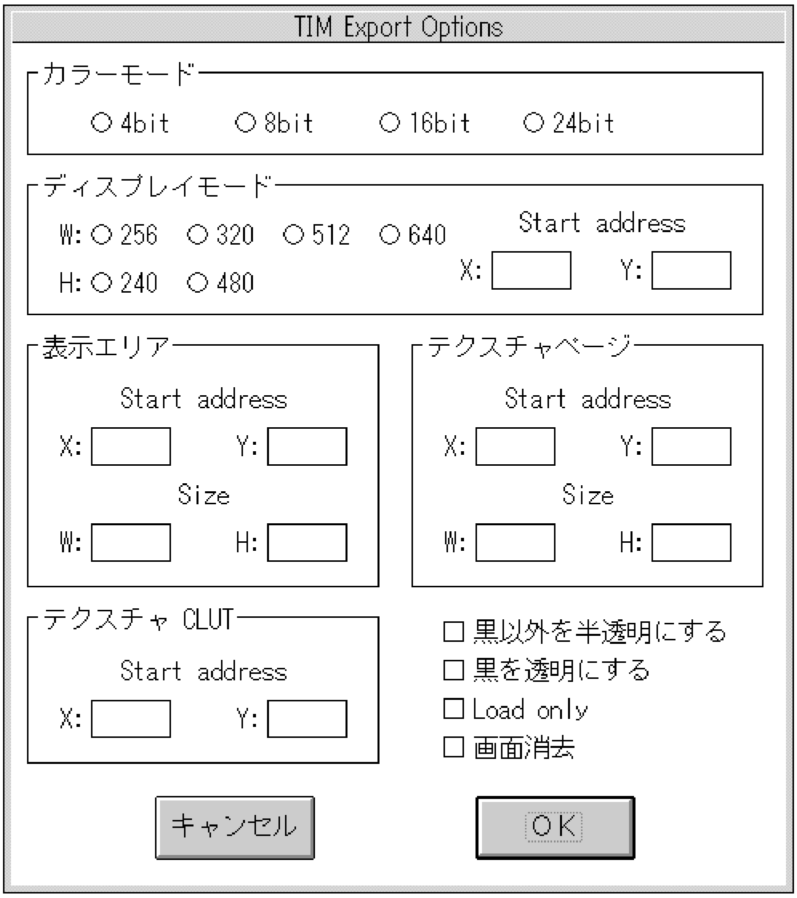


Figure 1-4-1 TIM Export Options dialog box

2. Set or select any desired value for the following settings and click “OK”.

- **Color mode**

Set a DTL-H201A pixel format. One of the following formats is available, depending on the Adobe Photoshop image mode.

Index color → 4 or 8-bit CLUT

RGB color → 16 or 24-bit Direct

- **Display mode**

Set the DTL-H201A horizontal (W) and vertical (H) resolutions. Change of the start address allows you to begin with any VRAM address you want to display (left top corner)

- **Display area**

Set a rectangular VRAM address (left top corner) to map the texture data and the width and height.
This is invalid in Load only mode.

- **Texture page**

Set a VRAM address (left top corner) to store the texture data and the width and height.
In Load only mode, the address is automatically set to (0,0).

- **Texture CLUT**

Set a VRAM address (left top corner) to place the CLUT. This is invalid when the color mode is 16/24-bit.

- **Letting any other color than the black turn semitransparent**

In order to let a color turn semitransparent, set an Adobe Photoshop format pixel which has a value of 8 or more for at least one of R, G and B or transparent control bit which is CULT entry.

- **Letting the black turn transparent**

In order to let the black turn transparent, clear any Adobe Photoshop format pixel which has values of 7 or less for all of R, G and B or transparent control bit which is CULT entry.

- **Load only**

Changes the system mode to Load only mode. This mode merely sends the texture data to the VRAM. It does not map it into a rectangle of display area. Therefore, the transparent control is invalid.

Selection of the 24-bit color mode sets automatically this mode.

- **Screen clear**

Clears the whole VRAM before drawing.

- **Cancel**

Cancels drawing.

- **OK**

Outputs the drawing command to DTL-H201A.

4. NOTE

- In the Adobe Photoshop format, components R, G and B have 8-bit resolution. On the other hand, each resolution of DTL-H201A is 5-bit, except 24-bit mode. Therefore, outputted to DTL-H201A, 5 high order bits are valid but 3 low order bits are ignored unconditionally.

  For example

	Adobe Photoshop		DTL-H201A internal notation
(R, G, B):	(0x08, 0x07, 0xff)	→	[0x01, 0x00, 0x1f]

- DTL-H201A is provided with the transparent control bit (except 24-bit mode) but Adobe Photoshop has nothing which corresponds to it. This module is provided with 2 types of check box to deal with the transparent control bit. These boxes allow you to use the transparent control bit to some degree. However, “transparent” and “(semi) nontransparent black” cannot be mapped into a display area at a time. In such case, use (R, G, B)=(0, 0, 8) and others instead of “black”.

- The transparent control bit is valid only when texture is mapped into the display area.

5 TIMFMT.8BI

1. General

Features

TIMFMT.8BI is a plug-in module which allows you to import directly the two dimensional TIM format file that the PlayStation system deals with from Adobe Photoshop and output image which was created with Adobe Photoshop as the TIM format file.

This software has the following features.

- **Supports 4 TIM format modes**

  The software supports all image data modes (4/8-bit CLUT and 16/24-bit Direct) that PlayStation system can deal with.

- **Supports the transparent control bit (partly)**

  The software allows you to use the transparent control bit (STP) to save the image data as the TIM format file.

- **Allows you to specify the VRAM load address**

  The software allows you to specify Texture CLUT and Texture page addresses to load the image data which was saved as the TIM format file to the VRAM.

Operating requirement

TIMFMT.8BI is subject to the Adobe Photoshop operating requirement because it is a module which is started up from Adobe Photoshop. General matters of Adobe Photoshop are not explained here because this manual is written for those who are familiar to its operating procedure. For details of Adobe Photoshop, refer to instruction manuals and others attached to Adobe Photoshop.

System configuration

In order to use TIMFMT.8BI, the following software is required.

- Adobe Photoshop 2.5J or more

2. Preparation

Installation of program

Copy TIMFMT.8BI at C:\PSXGRAPH\BIN to PLUGINS subdirectory in a directory in which PHOTOSHP.EXE was installed.

3. Operations

In order to open a TIM format file as an Adobe Photoshop document

"Open..." command is available to open a document when the file extension of "*.TIM" is attached to a TIM format file you want to open. Otherwise, use "Open in specified format..." command unless the file extension of "*.TIM" is attached to.

■ How to use "Open..." command

1. Select "Open..." option from the file menu to access "Open" dialog box.

2. Select a TIM format file you want to open and click "OK".

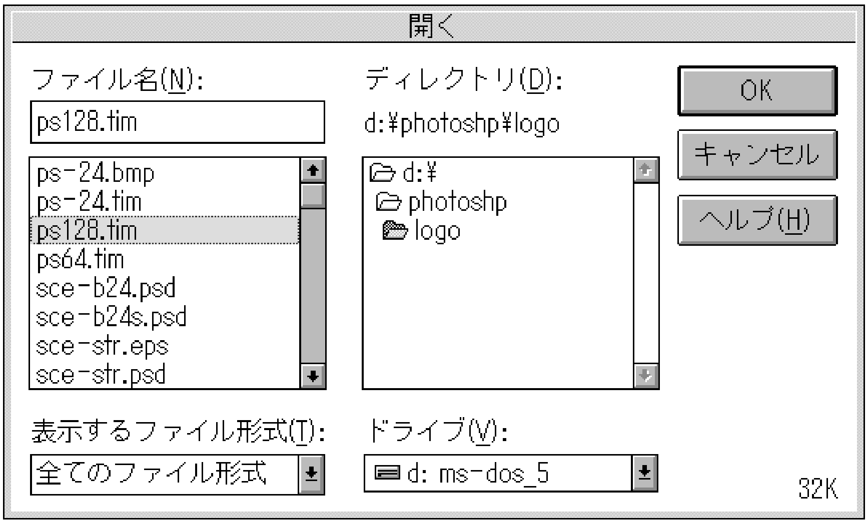


Figure 1-5-1 "Open" dialog box

■ **How to use “Open in specified format...” command**

1. Select “Open in specified format...” option from the file menu to access “Open in specified format” dialog box.

2. Select “TIM (*.TIM)” option from the “File format” drop-down list.

3. Select a TIM format file you want to open and click “OK”.

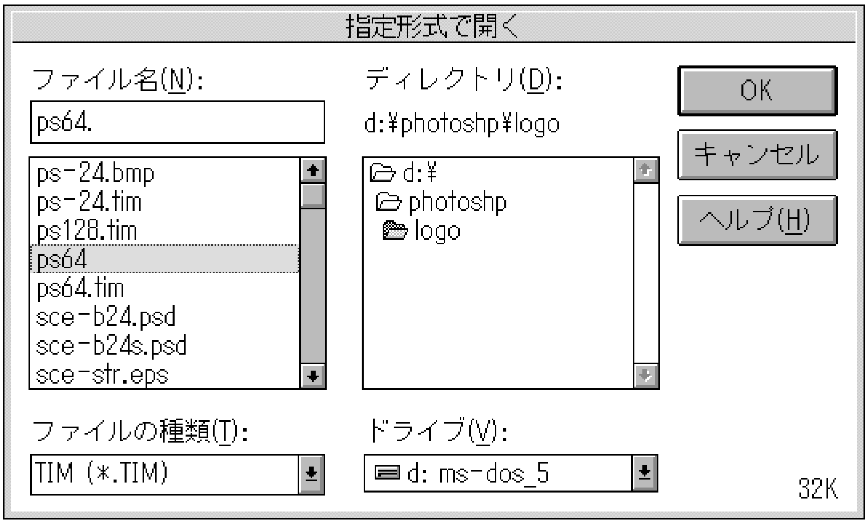


Figure 1-5-2 “Open in specified format” dialog box

In order to save the Adobe Photoshop image as the TIM format file

In order to save a file which has any other format than the TIM format as the TIM format file, use “Save with another name...” command.

■ **How to use “Save with another name...” command**

1. Select “Save with another name...” option from the file menu to access “Save with another name” dialog box.

2. Select “TIM (*.TIM)” option from the “File format to save” drop-down list.

3. Enter any desired filename and click “OK”.

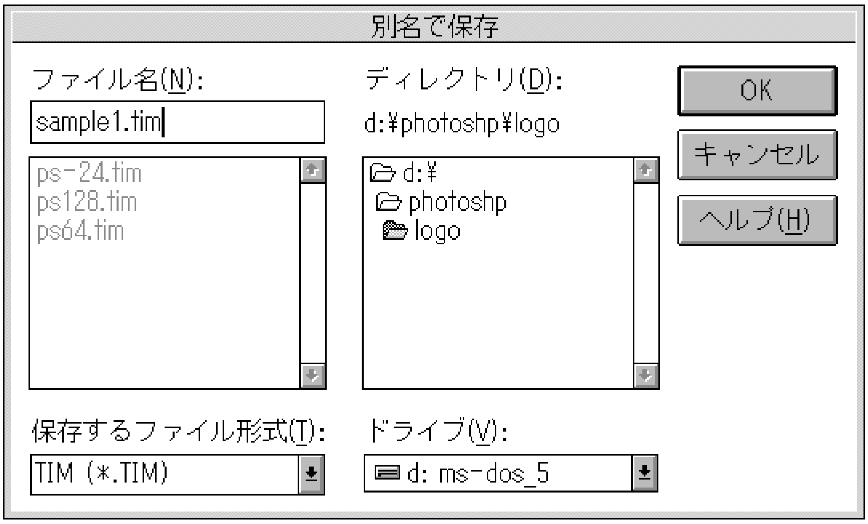


Figure 1-5-3 "Save with another name" dialog box

4. A warning "Image information which cannot be saved exists. Save?" appears.

Click "OK".

Figure 1-5-4 Warning about image information which cannot be saved

5. TIM Format Output Options dialog box appears. The Texture CLUT address and Texture page address show values of a TIM format file which was opened (or saved) at the previous time.

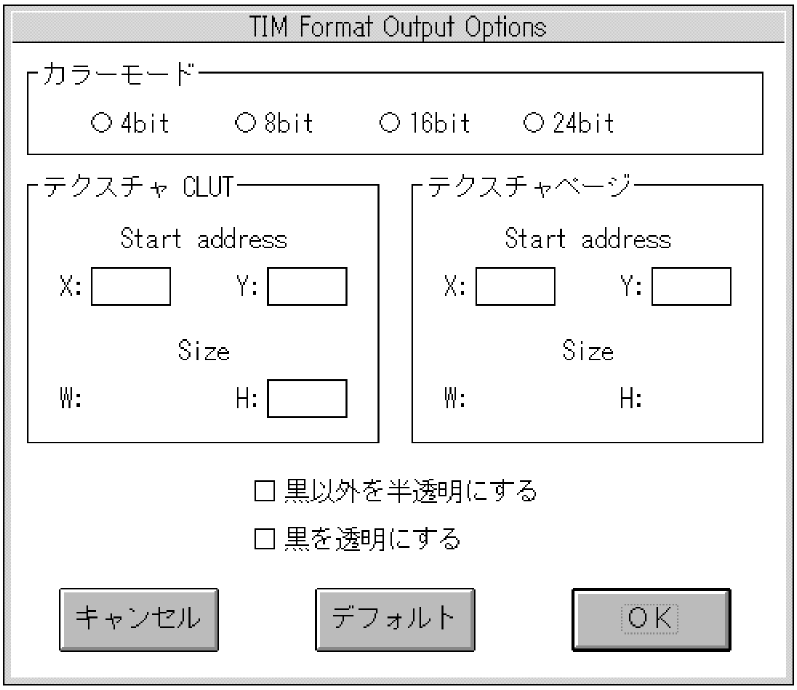


Figure 1-5-5 TIM Format Output Options dialog box

6. Set or select any desired value for the following settings and click “OK”.

- **Image data mode**

  Set PMODE of the TIM format file. One of the following formats is available, depending on the type of original image.

  Index color → 4 or 8-bit CLUT

  RGB color → 16 or 24-bit Direct

- **CLUT section**

  Set a VRAM CLUT address (left top corner). You can set the width (W) and height (H) of CLUT as follows.

  4-bit CLUT → Width: 16 (fixed), Height: 1-16 (as necessary)

  8-bit CLUT → Width: 256 (fixed), Height: 1 (fixed)

- **Pixel data section**

Set a VRAM pixel data address (left top corner). The width (W) and height (H) of image are shown but unchangeable.

- **Letting any other color than the black turn semitransparent**

In order to let a color turn semitransparent only in semitransparent mode, set an Adobe Photoshop format pixel which has a value of 8 or more for at least one of R, G and B or transparent control bit which is CULT entry. (The default turns this check box OFF.)

- **Letting the black turn transparent**

In order to let the black turn always transparent regardless of semitransparent mode, clear any Adobe Photoshop format pixel which has values of 7 or less for all of R, G and B or transparent control bit which is CULT entry. (The default turns this check box ON.)

- **Default**

Sets standard values for the CLUT section address and pixel section address. Undo of this operation is not allowed.

- **Cancel**

Cancels saving.

- **OK**

Saves the image information as the TIM format file.

4. NOTE

- In Adobe Photoshop, components, R, G and B have the 8-bit resolution. On the other hand, each TIM format resolution is 5-bit, except 24-bit mode. Therefore, to say strictly, equivalent conversion from Adove Photoshop to the TIM format file is not realized. (The 24-bit mode realizes it.) This module adds '0's for 3 low order bits when it opens the TIM format file and picks up 5 high order bits but ignores 3 low order bits when it saves the file.

  For example,

  Adobe Photoshop TIM format file Adobe Photoshop

  (R,G,B):(0x08,0x07,0xff) → (0x01,0x00,0x1f) → (0x08,0x00,0xf8)

- The TIM format is provided with the transparent control bit, except 24-bit mode but Adobe Photoshop has nothing which corresponds to it. Therefore, this module deals with the transparent control bit as follows.

  - **When it opens the TIM file**

    All transparent bits are lost. This leads both "transparent" and "(semi) nontransparent black" to the same Adobe Photoshop value, (R, G, B)=(0, 0, 0).

  - **When it saves a file as the TIM format file**

    Two types of check box are provided to deal with the transparent control bit. These boxes allow you to use the transparent control bit to some degree. However, you cannot store "transparent" and "(semi) nontransparent black" in a TIM format file at a time. In such case, use (R, G, B)=(0, 0, 8) (equal to [0, 0, 1] of the TIM file) instead of "black".

- Adobe Photoshop index color image holds always the 256-color CLUT. For 4-bit mode TIM format file, input and output of the CLUT of up to 256 colors are available. However, Adobe Photoshop uses only the first 16 colors to display it.

6 TIMPOS.EXE

1. General

This software changes the VRAM image and pallet (CLUT) addresses stored in the PlayStation image data file (TIM).

2. How to use

TIMPOS [options] input-file image-x image-y [clut-x clut-y]

For TIM file given by the argument, set the VRAM addresses to which the image and pallet (CLUT) are located. Addresses which you set are coordinate values of the top of the left side of the image and pallet. The VRAM space is 1024 wide x 512 long. Use of the argument does not allow you to set only the pallet VRAM address. Set the image VRAM address at the same time. If you enter a minus number for any address, the address is never changed. This allows you to change only the pallet VRAM address (Example 1).

Omission of the extension ".TIM" of the TIM filename given by the argument is allowed.

3. Options

-o output-file

Specify an output filename. The default is a filename identical with the input filename.

-v

Outputs information about the input file and conversion to a standard output. Input of only the TIM file together with this option allows you to know the current address information.

4. Example

Example 1; Changing only pallet address

```
C:>TIMPOS TEST.TIM -1-1 0500
```

7 TIMUTIL.EXE (TIM UTILITY)

1. General

TIM utility is an Windows application which allows you to convert each bit map format of the PlayStation TIM, Windows BMP, Macintosh PICT and general-purpose RGB to one of them.

2. How to use

Selection of “Bit map file” sub-option from “Open...” option of the file menu lets you access the Parameter Setting window.Change properly parameter values and then press “Convert...” button. Or select “Give name to save...” option from the file name. File Save dialog pops up. Entering a post-conversion filename starts format conversion.

You can drag an input file from the File Manager to drop it in the TIM utility window instead of using “Open...” command.

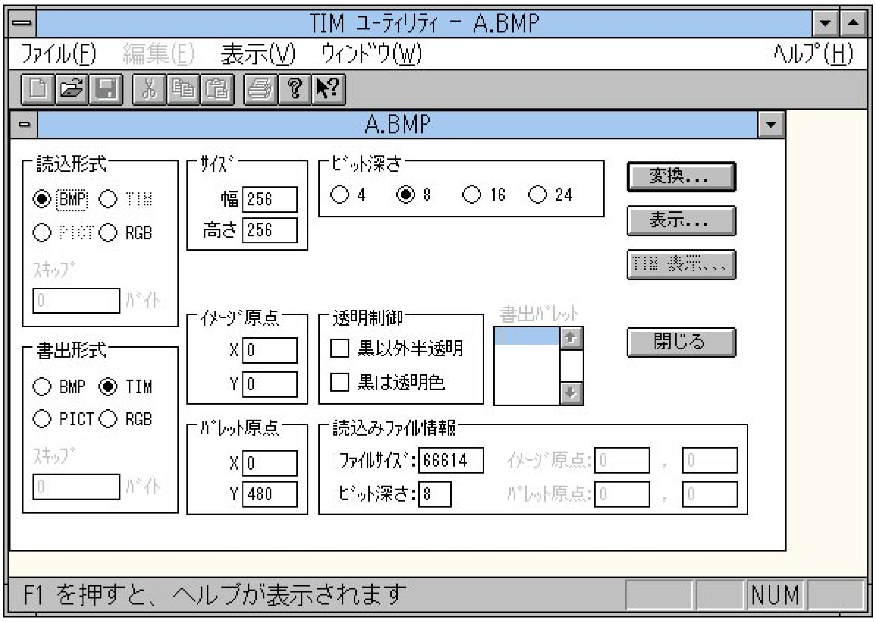


Figure 2-7-1 Parameter Setting window

3. Explanation of Parameter Setting window settings

Read-in format

Shows the format of a file which was read in. When it shows one of the TIM, BMP and PICT format, you can select the RGB format. Header information of the TIM, BMP or PICT is ignored during conversion and the file is compulsorily read in as the RGB data.

"TIM"

Is selected when a file which was read in can be constructed as the PlayStation TIM.

"BMP"

Is selected when a file which was read in can be constructed as the Windows BMP.

"PICT"

Is selected when a file which was read in can be constructed as the Macintosh PICT.

"RGB"

Is selected when a file which was read in cannot be constructed as one of the TIM, BMP and PICT.

"SKIP"

You can enter here when read-in format is the RGB.

As many bytes as you specify here will be skipped when an input file will be read in as the RGB format.

Write-out format

Select a post-conversion file format.

"TIM"

Converts a file into the PlayStation TIM.

"BMP"

Converts a file into the Windows BMP.

"PICT"

Converts a file into the Macintosh PICT.

"RGB"

Converts a file into the general-purpose RGB format.

Selection of this button allows you to change values for the SKIP of the write-out format.

"SKIP"

You can enter here when write-out format is the RGB.

As many 0s as number of bytes you specify here will be added to when a file will be written out as the RGB format.

Size

You can enter here when read-in format is the RGB. You must always enter here to construct byte string of the input file as the RGB format image data. If the image data size calculated from a value entered here and a value entered for the SKIP of the read-in format is larger than the size of a file which is inputted, it causes an error. If it is smaller, it lets a warning panel pop up.

Image origin

You can enter here when write-out format is the TIM. Set the PlayStation VRAM image origin coordinates which are inserted in the output TIM file.

Pallet origin

You can enter here when write-out format is the TIM and the bit depth is 8 or less. Set the PlayStation VRAM pallet (CLUT) origin coordinates which are inserted in the output TIM file.

Transparent control

You can set the transparent control when write-out format is the TIM and the bit depth is 16 or less.

Selection of "Any other color than the black is semitransparent" option sets the transparent control bit to 1 for any pallet entry which has any other value than 0 for one of R, G and B.

Selection of "Black is transparent" option sets the transparent control bit to 0 for any pallet entry which has 0s for all of R, G and B. Lack of this selection lets any color which has 0s for R, G and B turn nontransparent black.

Bit depth

Specify the number of bits per pixel of the output file.

When write-out format is the BMP, you can specify one of 4, 8 and 24-bit. When it is the PICT, you can specify one of 4, 8 and 16-bit. When it is the RGB, only 24-bit is available.

Write-out pallet

Select a pallet which is written out to a post-conversion file when read-in format is the TIM and multiple pallets are provided. Pallet which you select here will be also used in the Display and TIM display.

If you set “Reflect on origin” of “Write-out pallet” with “Set...” command of the file menu, Y coordinate value of the pallet origin increases or reduces automatically, depending on the place of the selected entry in the write-out pallet list.

Information about read-in file

Shows the read-in file size and pixel depth.

It shows “Image origin” when read-in format is the TIM. Moreover, it also shows the “Pallet origin” when the “bit depth” is 8 or less.

Conversion

Executes format conversion according to parameters you set. As the File Save dialog pops up, enter a post-conversion filename. This function is equal to “Give name to save...” option of the file menu.

Display

Reads in a file according to the current read-in format and displays it. Colors are approximated when the bit depth of the read-in file is larger than that of a display you use. (The current version may require much processing time.)

TIM display

Displays file contents on the Artist board when the current read-in format is the TIM. Pressing this button lets you access the following dialog.

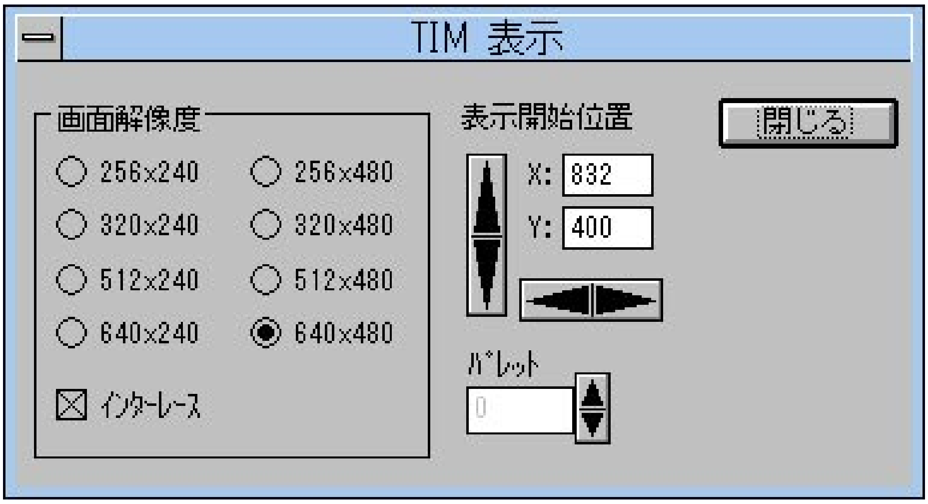


Figure 2-7-2 TIM Display dialog

This dialog allows you to set a screen resolution, a VRAM position corresponding to a video monitor screen connected to the Artist board and, in case of the TIM with multi pallets, a pallet used for the Display, while the TIM file image being displayed on the Artist board.

When the bit depth is 4 or 8, the TIM display displays file contents by texture mapping to a polygon. Moving the display area allows you to view directly 4 or 8-bit index data or the pallet area.

When the bit depth is 4 or 8, the TIM file image data is transmitted to the image origin, as a rule, but when X of the origin is below 640, it is compulsorily transmitted to (640,y). This prevents it from overlapping with the display area. The pallet data (CLUT) is transmitted to the pallet origin. When the bit depth is 16 or 24, the TIM display displays file contents by transmitting a bit pattern to the image origin.

The display area is initialized so that the image is situated at the center of the 640 x 480 display screen.

Close

Closes the current window.

This function is equal to “Close” of the file menu.

4. Menu bar

"Open..." command ([File] menu)

Opens existing bit map files. The following dialog box appears.

Figure 2-7-3 File Open dialog

This dialog allows you to set the following options.

Filename

Enter or select a bit map filename you want to open. List box shows files which have an extension corresponding to a file format you selected in the "File type" box.

File type

Select a file format of bit map file you want to open.

TIM utility supports the PlayStation TIM, non-compressed format Windows BMP, PICT which includes one or more bit maps, except 32-bit mode and general-purpose RGB. (Write-out bit depth is limited to 4, 8, 16 or 24-bit.)

You can also select the RSD format model data. This opens all of the TIM data which are specified by the model data.

Drive

Select a drive in which a bit map file you want to open will be saved.

Directory

Select a directory in which a bit map file you want to open will be saved.

"Close" command ([File] menu)

Closes a window which is currently active.

"Give name to save..." command ([File] menu)

Give a filename to a bit map file you are dealing with to save it. As the following dialog box appears, you can give a filename to a bit map file you are dealing with to save it. This executes format conversion according to parameters you set.

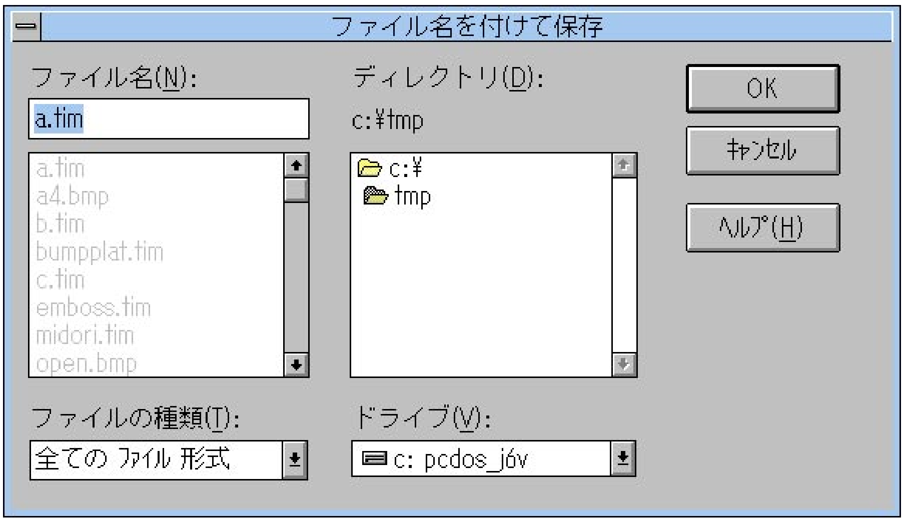


Figure 2-7-4 File Save dialog

This dialog box allows you to set the following options to specify a filename or location in which you save it.

Filename

As a rule, enter a name different from the original one. Give a filename of up to 8 half size characters and add an extension of 3 half size characters to it.

Drive

Select a drive in which you save a bit map.

Directory

Select a directory in which you save a bit map.

As a rule, overwriting to the read-in file is not allowed.

Only exception is the case in which both read-in format and write-out format are the TIM and no other setting than the image and pallet origins is changed. In this case, you can overwrite it. (This is equal to TIMPOS.EXE of the Comandline version.)

"Save all files..." command ([File] menu)

Saves all bit maps that you are dealing with. As the following dialog box appears, specify a directory in which you write out them. This executes format conversion according to parameters set in each bit map. Filenames are identical with the original ones except extensions which are replaced by an extension specified in the write-out format.

Figure 2-7-5 Directory Select dialog

This dialog allows you to set the following options to specify locations in which you save bit maps.

Drive

Select a drive in which you save bit maps.

Directory

Select a directory in which you save bit maps.

As a rule, overwriting to the read-in files is not allowed.

If any write-out file conflicts with the rule, it causes an error and no conversion is executed at all.

"Set..." command ([File] menu)

Sets parameter initial values which will be set in the Parameter Setting window when you will open a file.

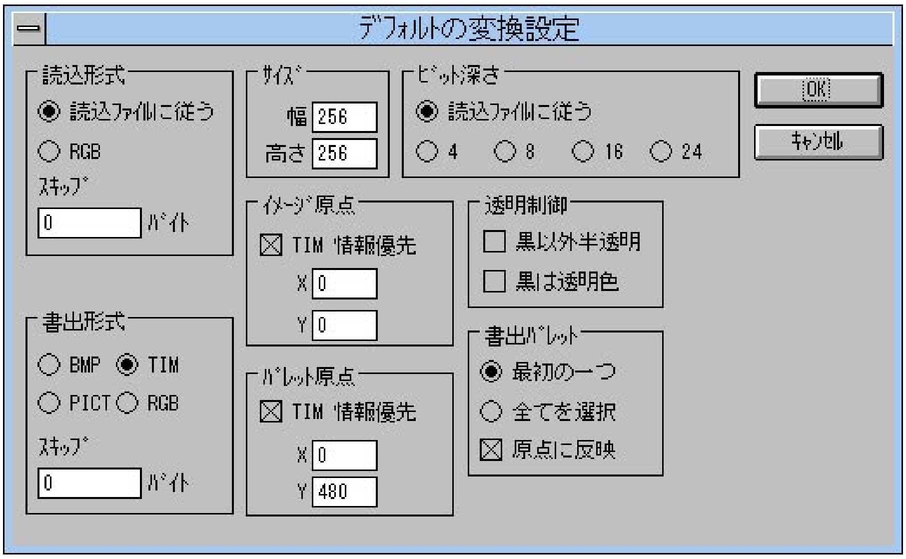


Figure 2-7-6 Set dialog

If "Subject to read-in file" is set in "Read-in format", a format which was judged from contents of the read-in file is set in the read-in format of the Parameter Setting window. If "RGB" is set, any file is set to the RGB. This also applies to the "Bit depth".

"Image origin" and "Pallet origin" are used as default values for conversion parameters when "TIM information priority" is not selected or when read-in format is any other format than the TIM.

For "Write-out pallet", you can select a write-out pallet from the write-out pallet list when a TIM file with multiple pallets (CLUT) is read in. Determine the default.

Selection of "First one" selects only No. 0 pallet. Selection of "Select all" selects all pallets as write-out pallet. The "Write-out pallet" is provided with another button, "Reflect on origin". This button sets whether the pallet origin is automatically set or not when an entry is selected from the write-out pallet list. If this is set, selection of No. 0 entry from the write-out pallet list sets the pallet origin which is set in the original TIM file for the pallet origin. Selection of No. 1 entry adds 1 to Y coordinate of the pallet origin which is set in the original TIM file and sets the value for the pallet origin.

Other parameter values than mentioned above are simply used as the initial values in the Parameter Setting window.

Values set in the dialog will be saved at end of TIM utility.

"TIM Locate..." command ([Window] menu)

Opens "TIM Locate" dialog to locate graphically a TIM format file which is opened in the VRAM.

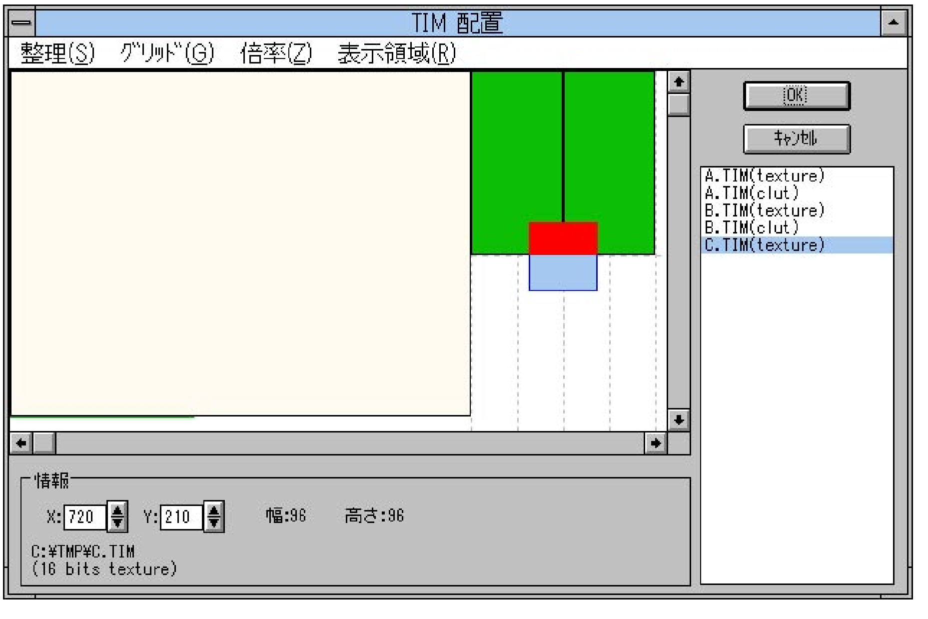


Figure 2-7-7 TIM Locate dialog

“TIM Locate” dialog has, to divide roughly, 4 main areas shown below.

(1) VRAM image area
(2) Information area
(3) Select list area
(4) Menu bar

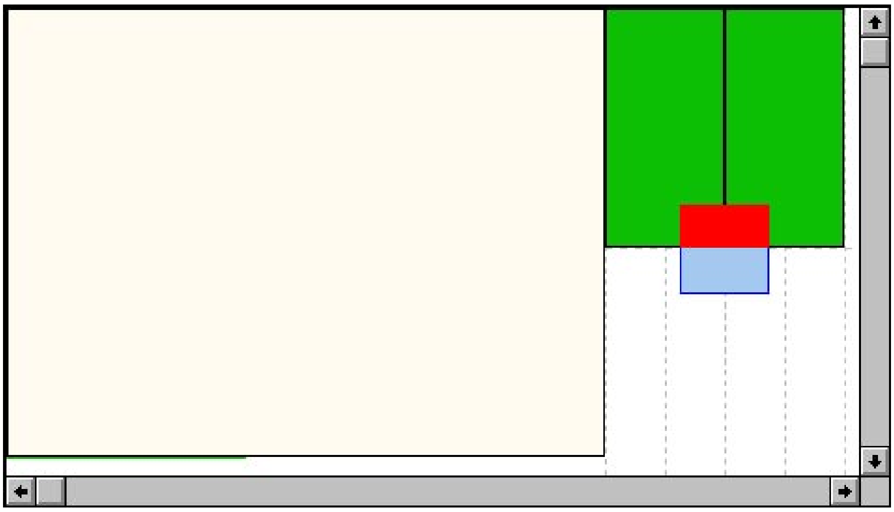


Figure 2-7-8 VRAM image area

This area displays the VRAM image on PlayStation. Rectangles which show image or pallet (CLUT) of the currently opened TIM file, a rectangle which shows a range which is displayed and texture page borderlines which are referred to for location are displayed. For image or pallet rectangle, the part which does not overlap with another image or pallet rectangle or the range rectangle is displayed in green and the part which overlaps with other rectangles or projects from the VRAM area is displayed in red. The left button of the mouse allows you to select these rectangles and you can move them by dragging. Use of the left button of the mouse while pressing the Shift key or Ctrl key allows you to select multiple image or pallet rectangles. Rectangles you select are displayed in blue.

Figure 2-7-9 Information area

This area displays information about selected image or pallet rectangle. Image or pallet origin coordinates, a VRAM range the rectangle occupies, a filename and the bit depth are displayed.

You can change the origin coordinates by clicking Up and Down arrow buttons or entering directly numeric values in the text part. This area displays nothing when multiple image or pallet rectangles are selected.

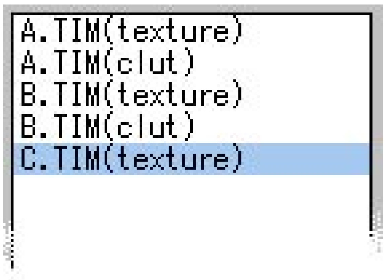


Figure 2-7-10 Select list area

The Select list area displays a list of images and pallets which can be dealt with. This list allows you to select image or pallet rectangles. Rectangle which was selected in the VRAM image area is regarded to be selected also in the list. Use of the Shift or Ctrl key allows you to select multiple rectangles. Dragging also allows selection of multiple rectangles.

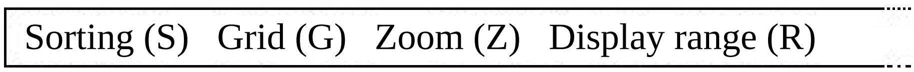


Figure 2-7-11 Menu bar

■ Sorting menu

"Texture"

Sets the left top corner of all textures to a side of the nearest texture area or display range. Selection of this menu after setting roughly textures with the mouse arranges them orderly.

"Pallet"

Paves all pallets vertically from the bottom of a range which is currently displayed.

■ Grid menu

"Non"

Disables grid function. Texture is located to coordinates you specified with the mouse.

"Texture page"

Moves and lines up textures along the nearest texture page borderline from coordinates you specified with the mouse.

Both X and Y coordinates are lined up along the texture page borderline if you selected "XY", only X coordinates are lined up if you selected "X" and only Y coordinates are lined up if you selected "Y".

"Magnet"

Moves and sets textures to a side of the nearest texture area or display range from coordinates you specified with the mouse. Both X and Y coordinates are lined up along the texture page borderline if you selected "XY", only X coordinates are lined up if you selected "X" and only Y coordinates are lined up if you selected "Y".

■ Zoom menu

Sets scales for the image display in the "VRAM image area". Use this for small texture, pallet and others when it is difficult to select or move them with ordinary scales.

■ Display range menu

Changes display resolutions which are actually used on PlayStation. Change of the display range changes the range of the display area.

Press "OK" button to reflect edited VRAM images on the Parameter Setting window. Selection of "Cancel" button or "Close" of the system menu invalidates move of image or pallet.

5. Remarks

- **The compressed format BMP, JPEG compressed PICT, 32-bit PICT and PICT without bit map cannot be read in.**

- **Write-out as the compressed format BMP or JPEG compressed format PICT file is not allowed.**

- **Write-out file bit depth is limited to 4, 8, 16 or 24-bit.**

- **In data conversion in which the bit depth reduces, for example, from 16-bit to 8-bit, color information is approximated. As this approximation is not realized with a method such as color compression but with color map on which R, G and B are uniformly allocated, the color accuracy may deteriorate.**

- **As a rule, overwriting to the file which was read in is not allowed.**

  However, overwriting is allowed when both read-in format and write-out format are the TIM and only image origin and pallet origin were changed.

  This function is equal to TIMPOS.EXE of Commandline version.

CHAPTER 2

3D UTILITY

1 DXF2RSD.EXE

1. General

This software converts the DXF file into the PlayStation 3D model data format (RSD).

2. How to use

DXF2RSD [options] DXF-files...

Input of a DXF file with the argument creates 4 files shown below.

- File-related information file (*.RSD)
- Polygon information file (*.PLY)
- Material information file (*.MAT)
- Group information file (*.GRP)
- These 4 files are put together into one and called “RSD” or
- “RSD data”. The wildcard is available for the argument. you can input and convert multiple files at a time. Omission of
- “.DXF” of the filename is allowed.

3. Options

-o output-file

Specify an output RSD name. The extension is removed. The default creates a filename identical with the input filename, except the extension which is removed.

-col r g b

Specify a color of the whole model with R, G and B (0-255, respectively). The default is the gray (200 200 200).

-cf color-file

Specify a color table file.

-cl

Outputs a list of undefined colors to a standard output. This sorts polygons of the same color and outputs them to a MAT file. The default is OFF. (Refer to example 2.)

-info

Shows information about the input DXF file. This lets you know the approximate size and number of polygons. No conversion is executed.

-max n-poly

Set the maximum number of polygons which can be converted. The default is 10000.

quad or -quad1

Triangulates no 4-vertex 3DFACE. This allows reduction of the number of polygons of the whole model. The default is OFF.

-quad2

Converts 2 adjacent triangles into one quadrangle as many as possible. This is not provided yet.

-quad3

Converts the triangle into the quadrangle whose third and fourth vertices are equal to each other. This allows any triangle to turn to a quadrangle.

-s or -g

Executes the smooth (Gouraud) shading. The Default is OFF.

-e distance

Regards all vertices which exist in a sphere having a radius of distance given with the argument as the same. This allows reduction of the numbers of vertices and polygons. Distance calculation is executed after scale-up or scale-down with the -sc option.

-r

Prevents overlapping normal vectors from being created to reduce the number of normals. This is useful to the flat shading. The default is OFF. (Refer to example 3.)

-n

Creates no normal. Set this when no light calculation is required. The default is OFF.

sc factor

Scales up or down the model. Specify a scale factor with the argument. The default is 1.0.

t x y z

Moves the model. Specify a movement with the argument. The default is (0.0, 0.0, 0.0).

auto

Moves the model near to the origin and scales up or down it (so that it can settle in a cube having a side of 1000.) The default is OFF.

back

Reverses the normal vector of all polygons. The default is OFF.

both

Creates both sided polygons. The default is OFF.

dup

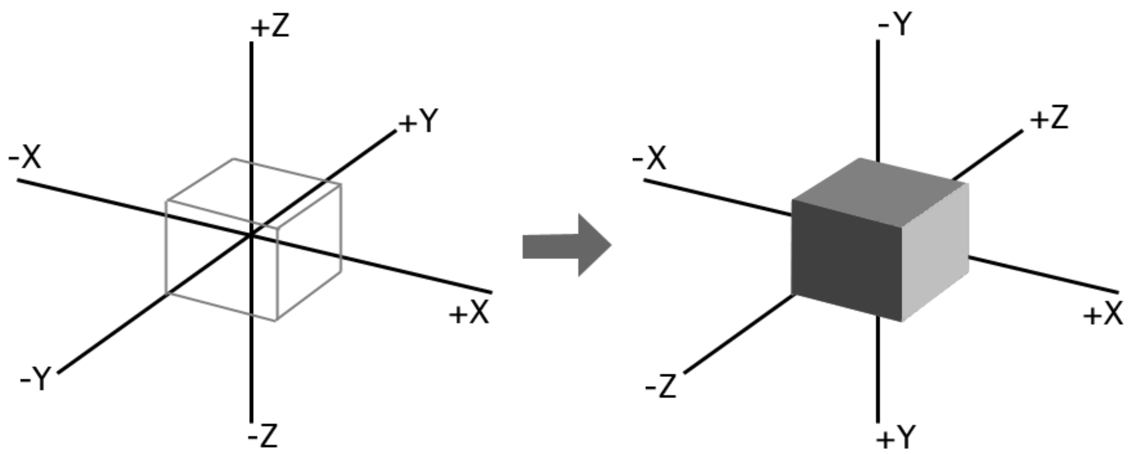
Creates both front and rear polygons of all polygons. This doubles the number of polygons. The default is OFF.

nopl

Converts only the 3DFACE, ignoring Polyline. The default is OFF.

Y-Z, -Y+Z, +Y-Z, +Y+Z, -Z-Y, -Z+Y, +Z-Y, +Z+Y

Specify a coordinate system conversion method. Specify the front and upper axes and their directions of a coordinate system of a modeler which is seen from the front. For example, “-Y+Z” means that the front axis is -Y axis and the upper axis is +Z axis. The coordinate system which is mentioned here is for the DXF and does not always correspond to the modeler screen. The default is “-Y+Z”. DXF2RSD converts it into the PlayStation coordinate system (-Z-Y). (In the PlayStation coordinate system, the front axis is -Z axis and the upper axis is -Y axis.)



Coordinate system of modeler (-Y+Z type) Coordinate system of PlayStation (-Z-Y type)

Figure 2-1-1 Conversion of coordinate systems

-v

Outputs detailed information about conversion to a standard output. (Refer to example 1 shown later.)

4. Constraint

The current version has the following constraints.

- Is compatible with only the ASCII format DXF.
- Is compatible with only the 3DFACE and Polyline of DXF entities.
- May not convert a large Polyline. Convert it to the 3DFACE (triangle or quadrangle) as many as possible with a modeler and then create the DXF file.
- The current -quads [1-3] do not couple 2 triangles to create a quadrangle. Quadrangle which does not have 4 vertices on the same plane may not be displayed correctly.
- Numbers of vertices and normals to be created affect the maximum number of polygons which can be converted.Generally, it is about 5000 polygons.

5. Supplement

- Both the 3DFACE and Polyline are DXF polygon representation formats. The 3DFACE represents a polygon (triangle or quadrangle) with 4 vertices. On the other hand, the Polyline represents multiple polygons with coupled segments. The 3DFCE is apt to cause data to be larger but has the superior compatibility. The Polyline allows data to be smaller but data conversion between different modelers may not be realized correctly.

  The 3DFACE can be converted directly into the RSD but the Polyline requires triangulation before conversion. The triangulation costs much and is not always successful. (An error message “Fail to triangulate!” appears.) Even if the triangulation is completed successfully, directions of some polygons may be reversed.

  However, a Polyline which 3D studio and others create is called POLYFACE MESH and equal to the 3DFACE representation. Therefore, there is no problem about conversion.

- A practical number of polygons which can be moved as an object on PlayStation is about 2000 at maximum. Create a model data, taking into consideration of it.

- Selection of the Gouraud shading may allow conversion even if conversion with the flat shading is impossible because of too many polygons.

- Every exchange of Y and Z coordinate axes or of plus and minus signs of each axis reverses the front and back of the polygon.

- For some modelers, even the 3DFACE output may reverse the front and back of the polygon. If the whole polygon is reversed, change the coordinate system (for example, +Y+Z) or use the - back option. If it is reversed partly, reversion of the part with the modeler or material editor is required.

● Commercial 3D modelers which are found to be convertible with DXF2RSD are shown below. Options for coordinate system conversion are also given.

3D modeler (version)	Platform	Coordinate conversion option
form Z (2.1 and later)	Mac	-Y+Z
Strata Studio Pro (1.0)	Mac	+Z+Y
StrataVision	Mac	+Z+Y
Sculptura (1.1)	PC	-Y+Z
trueSpace (1.0)	PC	-Y-Z
Alias Upfront (1.1)	PC	-Y+Z
3D Studio	PC	-back

Commercial 3D modelers which are found to be unconvertible now are as follows.

MacroModel (1.5)	PC
ShadeIII (1.0)	Mac

The sufficient condition for “To be convertible” is that it can output the 3DFACE DXF. Some are convertible even if they do not satisfy the condition. Moreover, some which cannot be converted directly may become convertible when you read in and save the DXF once with another modeler.

● This software creates a RSD without normal, regarding that the -n option is specified unless a too large data allows creation of all normals.

● The -r option is invalid in the Gouraud shading. You cannot use the -r option in the normal MIMe animation because the number and order of normals are changed if the -r option is specified.

6. Example

Example 1: Output when the -v option is used

```
C:> DXF2RSD -v -auto +Z+Y -quad -s FOO
```

Å@Å@Å@Å@Å@Å@Å@Output	Description
`Input DXF file: FOO.dxf`	Input DXF filename
`[DXF] SIZE    : 40230 lines`	DXF filename
`VERTEX    : 4320`	Number of vertices in the DXF
`POLYGON  : 1468 (estimate)`	Estimated number of polygons
`Å@Å@Å@Å@Å@Å@ 3-poly     : 1376`	(Breakdown) Triangle 1376
`Å@Å@Å@Å@Å@Å@ 4-poly     :   32`	Quadrangle 32
`Å@Å@Å@Å@Å@Å@ (9<)-poly  :    2`	Decagon and more 2
`Å@Å@Å@Å@Å@Å@ polylines  :    2 (max size=32)`	Polyline 2 (32 vertices)
`RANGE  x:   -1.015 ...   +0.785`	Minimum value...Maximum value... of each axis
`Å@Å@Å@Å@    y:   -2.533 ...   +0.768`	(Y and Z axes have been converted into
`Å@Å@Å@Å@    z:   -1.161 ...   +0.625`	ones for PlayStation as necessary)
`SCALE   :  302.870`	Scale factor
`MOVE    : (dx,dy,dz)=( 34.788, 267.255, 81.207)`	Movement
`MATERIAL:    0`	Number of polygons which are colored
`[RSD] VERTEX  :  796`	Number of vertices after conversion
`POLYGON : 1468`	Number of polygons after conversion
`Å@Å@triangles  : 1436`	(Breakdown) Triangle 1436
`Å@Å@quadrangles:   32`	Quadrangle 32
`RANGE  x: -272.477 ... +272.477`	
`Å@Å@Å@Å@    y: -500.000 ... +500.000`	Minimum value... and Maximum value of each
`Å@Å@Å@Å@    z: -270.510 ... +270.510`	axis after scale-up/down and move
`MATERIAL:    0`	Number of polygons which have material
`NORMAL  :  796`	Number of normals
`Output files  : FOO.[rsd, ply, mat, grp]`	Output filename

Example 2: Use of color information

You can reflect color information you entered with a modeler on the RSD. Specify the -cl option. Polygons will be “reasonably” colored. The word “reasonably” means that a color which may differ from the color on the modeler is allocated to parts which had the same color. This is because the DXF holds only “color number” but the RGB value. Then, you can specify a color by editing the MAT file with a text editor or creating a color table file as follows.

```
C:> DXF2RSD -cl FOO > FOO.CL

C:> TYPE FOO.CL

      183
      40
      253
      0
      8
```

Enter a color which you want to allocate to each color number just after the color number with R, G and B values (0-255, respectively) by the text editor.

```
C:> TYPE FOO.CL

      183  100  100  200
      40   58   20   43
      253  10   100  10
      0    212  20   100
      8    0    128  126
```

(As material files, FOO.MAT which are sorted by color number have been created, you may edit directly the RGB field of the file. For MAT file specification, refer to “Appendix: File format RSD” or RSD.TXT.)

Specify the -cf option to start up DXF2RSD again.

```
C:> DXF2RSD -cf FOO.CL FOO
```

Colors which were specified in the color table file, FOO.CL are allocated to a newly created RSD. (The FOO.MAT is updated.)

(If you edit directly the FOO.MAT, restart-up of DXF2RSD is not required.)

Example 3: Conversion of large data

When a too detailed data exists, you can use the -e option to put together several vertices into one. In the following example, numbers of vertices, polygons and normals are reduced by regarding 2 vertices which are less 100 away from each other as a vertex. (Note that distance calculation is based on a scale after scale-up.)

Though depending on data, you can reduce the quantity of data without changing practically the form if you set a proper distance.

```
C:> DXF2RSD -v -e 100 -sc 1000 BIG.DXF
```

Output	Description
`Input DXF file : BIG.DXF`	
`[DXF] SIZE      : 134628 lines`	
`      VERTEX    : 18982`	
`      POLYGON   : 8618 (estimate)`	
`                  3-poly  : 1746`	
`                  4-poly  : 3436`	
`       RANGE  x :  -1.644 ... +1.545`	
`              y :  -2.352 ... +0.000`	Scale up it by 1000 times because it is too small
`              z :  -3.649 ... +3.993`	
`       SCALE    : 1000.000`	
`       MATERIAL :    0`	
`[RSD]  VERTEX   : 1208`	
`       POLYGON  : 2708 (68% reduced)`	Number of polygons reduces to a third
`                  triangles  : 2708`	
`       RANGE  x : -1643.811 ... +1545.072`	
`              y : -2352.365 ...    +0.000`	
`              z : -3649.154 ... +3992.687`	
`       MATERIAL :    0`	
`       NORMA    : 2708`	
`Output files : BIG.[rsd, ply, mat, grp]`	

Moreover, you can reduce the number of normals if you set the -r option.

2 DXF2RSDW.EXE

1. General

DXF2RSDW.EXE is a Windows application which allows you to convert the DXF format file which various modelers output into the PlayStation RSD format. Input of a DXF file creates four files, RSD file (*.RSD), Polygon file (*.PLY), Material file (*.MAT) and Group file (*.GRP).

2. How to use

Selection of a DXF file with “Open...” option of the “File” menu lets you access the Parameter Setting window. Change properly parameter values and then press the “Convert...” button or select “Give name to save...” option from the filename. The File Save dialog pops up. Entering a post-conversion filename starts format conversion.

You can drag an input file from the File manager to drop it in the DXF2RSDW.EXT window instead of using the “Open...” command.

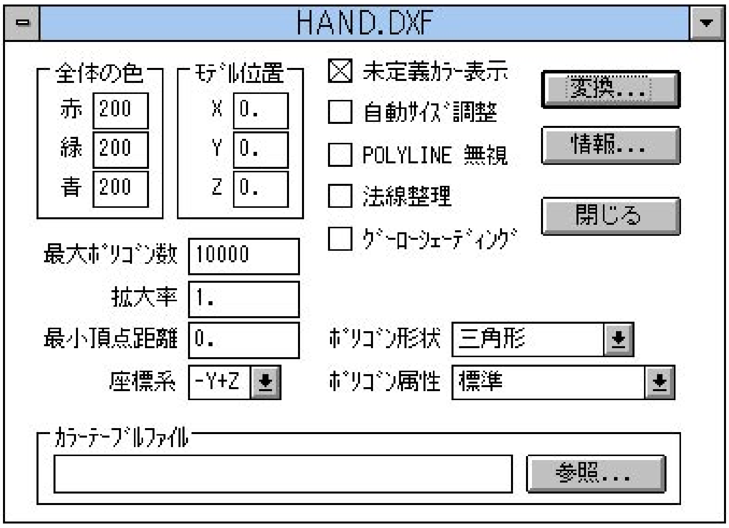


Figure 2-2-1 Parameter Setting window

3. Explanation of Parameter Setting window settings

Options corresponding to DXF2RSD.EXE of the Commandline version are also mentioned.

Color of the whole

Specify a color of the whole model with R, G and B values (0-255, respectively). This corresponds to the -col r g b option of the Commandline version.

Model location

Moves the model. Specify a movement. This corresponds to the -t x y z option of the Commandline version.

Maximum number of polygons

Set the maximum number of polygons which can be converted.
This corresponds to the -max n-poly option of the Commandline version.

Scale factor

Scales up or down the model. Specify a scale factor with a real number. This corresponds to the -sc factor of the Commandline version.

Minimum vertex distance

Regards all vertices which exist in a sphere having a radius of distance given here as the same. This allows reduction of the numbers of vertices and polygons. Distance calculation is executed after scale-up or scale-down with a scale factor. This corresponds to the -distance option of the Commandline version.

Coordinate system

Specify a coordinate system conversion method. Specify the front and upper axes and their directions of the coordinate system which is seen from the front. For example, “-Y+Z” means that the front axis is -Y axis and the upper axis is +Z axis. The coordinate system which is mentioned here is for the DXF and does not always correspond to the modeler screen. DXF2RSD converts it into the PlayStation coordinate system (-Z-Y). (In the PlayStation coordinate system, the front axis is -Z axis and the upper axis is -Y axis.)

This corresponds to various coordinate system options of the Commandline version.

Display of undefined colors

This corresponds to the -cl option of the Commandline version.
The default of this option differs from that of the Commandline version and is ON.

Automatic size adjustment

Moves the model near to the origin and scales up or down it (so that it can settle in a cube having a side of 1000.)
This corresponds to the -auto option of the Commandline version.

Polyline ignore

Ignores the Polyline and converts only the 3DFACE. This corresponds to the -nopl option of the Commandline version.

Normal reduction

Prevents overlapping normal vectors being from created to reduce the number of normals. This is particularly useful to the flat shading. This corresponds to the -r option of the Commandline version.

Gouraud shading

Executes the smooth (Gouraud) shading. This corresponds to the -s or -g option of the Commandline version.

Various polygons

There are options shown below.

Triangle

Triangulates all polygons.

Triangulation prohibit

Triangulates no 4-vertex 3 DFACE. This allows reduction of the number of polygons of the whole model. This corresponds to the -quad option of the Commandline version.

Quadrangle

Converts the triangle into the quadrangle whose third and fourth vertices are equal to each other. This corresponds to the -quad3 option of the Commandline version.

Polygon attributes

There are options shown below.

Standard

Creates polygons and normals according to information in the DXF file.

Normal reverse

Reverses normal vectors of all polygons. This corresponds to the -back option of the Commandline version.

Both sided polygon

Converts all polygons into both sided polygons. This corresponds to the -both option of the Commandline version.

Duplicate polygons

Creates both front and rear polygons of each polygon. This corresponds to the -dup option of the Commandline version.

Create no normal

Creates no normal. Specify this when no light calculation is required. This corresponds to the -n option of the Commandline version.

Color table file

Specify a color table file. This corresponds to the -cf color-file option of the Commandline version.

Pressing the “Refer...” button lets the File dialog pop up.

This allows you to enter a filename.

Conversion

Executes format conversion according to specified parameters. As the File Save dialog pops up, specify a post-conversion filename.

This function is equal to “Give name to save...” option of the file menu.

Information

Shows information about the input DXF file. You can know the approximate size and number of polygons.

This corresponds to the -info option of the Commandline version.

They are shown in a dialog shown below.

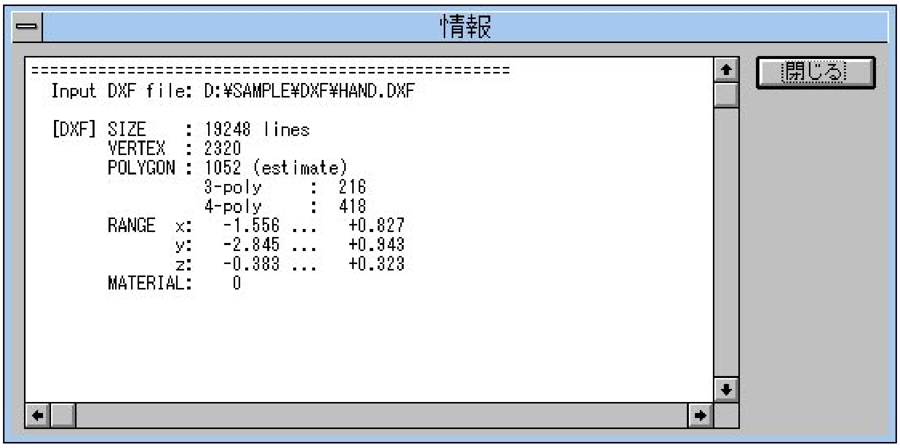


Figure 2-2-2 Information dialog

Selection of the text area with the mouse and pressing Ctrl + C copies them to the Windows clip board. To save them in a file and others, paste them with a proper text editor or others.

Close

Closes the current window. This function is equal to “Close” option of the file menu.

4. Menu bar

“Open...” command ([File] menu)

Opens existing DXF files. The following dialog box appears.

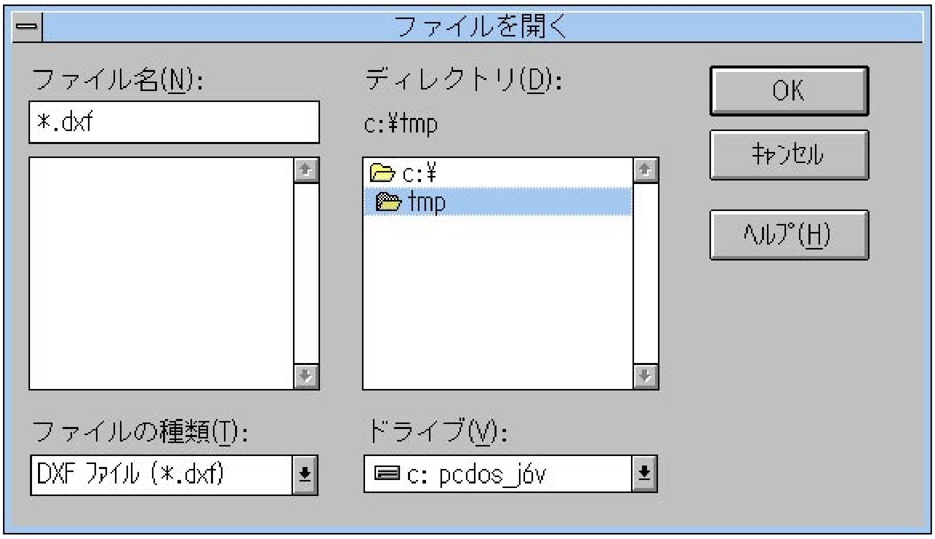


Figure 2-2-3 File Open dialog

This dialog allows you to set the following options.

Filename

Enter or select a DXF filename you want to open. The List box shows files which have an extension corresponding to a file format you selected in the “File type” box.

File type

Set whether you want to filter the file list with the *.dxf pattern or not.

Drive

Select a drive in which a DXF file you want to open is saved.

Directory

Select a directory in which a DXF you want to open is saved.

“Give name to save...” command ([File] menu)

Give a filename to a DXF file you are dealing with to save it. As the following dialog box appears, you can give a filename you want to a DXF file you are dealing with to save it. This executes format conversion to the RSD format according to parameters you set.

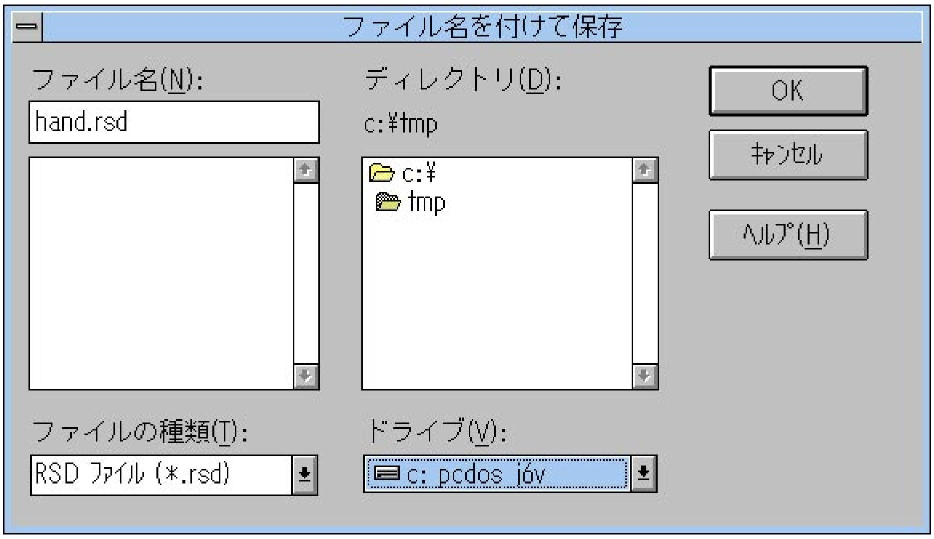


Figure 2-2-4 File Save dialog

This dialog box allows you to set the following options to specify a filename or location in which you save it.

Filename

Enter a filename different from the original name. Give a filename of up to 8 half size characters and add an extension of 3 half size characters to it.

Drive

Select a drive in which you save the RSD file.

Directory

Select a directory in which you save the RSD file.

5. Constraint

The current version has the following constraints.

- Is compatible with only the ASCII format DXF file.
- Is compatible with only the 3DFACE and Polyline of the DXF entities.
- May not convert a large POLYLINE. Convert it to the 3DFACE (triangle or quadrangle) with a modeler as many as possible and then create the DXF file.
- “Triangulation prohibit” never couples 2 triangles to create a quadrangle. Quadrangle which does not have 4 vertices on the same plane may not be displayed correctly.
- Numbers of vertices and normals to be created affect the maximum number of polygons which can be converted. Generally, it is about 5000 polygons.
- Use of color information

You can reflect color information you entered with the modeler on the RSD. Specify “Display undefined colors” option. Polygons will be “reasonably” colored. The word “reasonably” means that a color which may differ from the color on the modeler is allocated to parts which had the same color. This is because the DXF holds only “color number” but the RGB value. Then, you can edit the MAT file with a text editor or specify a color by creating a color table file with the “Display undefined colors” option. In this case,

“Undefined color” dialog pops up after completion of conversion from the DXF to the RSD.

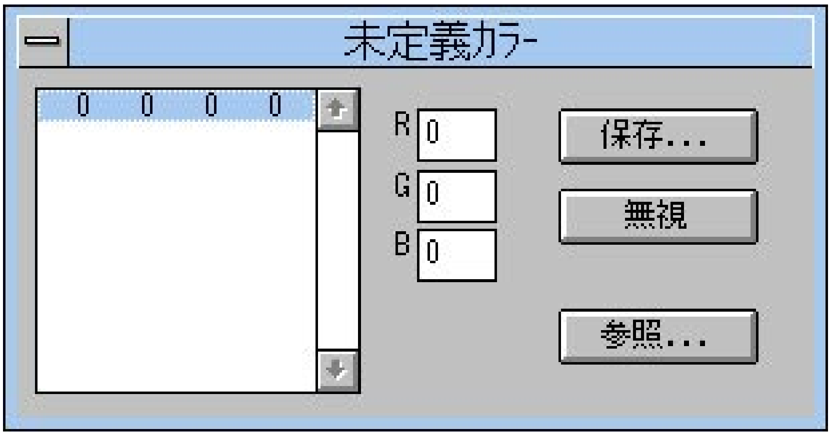


Figure 2-2-5 Undefined Color dialog

This dialog shows a list of undefined colors. Edit R, G and B fields and then press the “Save...” button. You can give a proper filename with the File dialog to save the list of undefined colors in a file. And then, specify the undefined color file, which you saved just before, in the “Color table file” option to “Convert...” again.

(If the “Display undefined color” is left specified, the “Undefined color” dialog appears again. Press “Ignore”.) Colors are allocated according to the color table file in a newly created RSD.

3 MIMEFILT.EXE

1. General

This software reads in multiple TMD files and creates the MIMe difference file.

2. How to use

MIMEFILT. [options] BASE.TMD VARIANT1.TMD VARIANT2.TMD...

Input of multiple TMD formats examines differences between the first TMD and the second and following TMDs to collect them in a difference file.

Differences between corresponding vertex coordinates and between normal vectors can be examined. The default creates only the vertex difference file (.VDF).

The wildcard is available for the argument. Omission of ".TMD" of the filename is allowed.

3. Options

-o output-file

Specify an output filename. This filename is used for the vertex difference filename (.VDF) and normal difference filename (.NDF). The default gives a filename identical with the original TMD filename except the extension which is removed.

-n

Creates a normal difference file (.NDF). The default is OFF.

-v

Outputs detailed information about conversion to a standard output. The default is OFF.

4. Constraints

- Numbers of vertices of all argument TMD files must be equal to each other. To create each TMD file, take care not to change the number and order of vertices. Create a basic prototype model with a 3D modeler which consists of sufficiently detailed polygons then move planes and vertices of the model to create a deformation model. (The number and order of vertices are not changed.) Save each model in the DXF format and convert it into the RSD format with DXF2RSD and then apply RSDLINK to it separately to create the TMD file.

5. File format

Format of the difference file created by MIMEFILT is the binary format in which blocks having the header and difference information for each object are coupled. (The normal difference file has the same format.)

```
C:> MIMEFILT MO.TMD M1.TMD M2.TMD
```

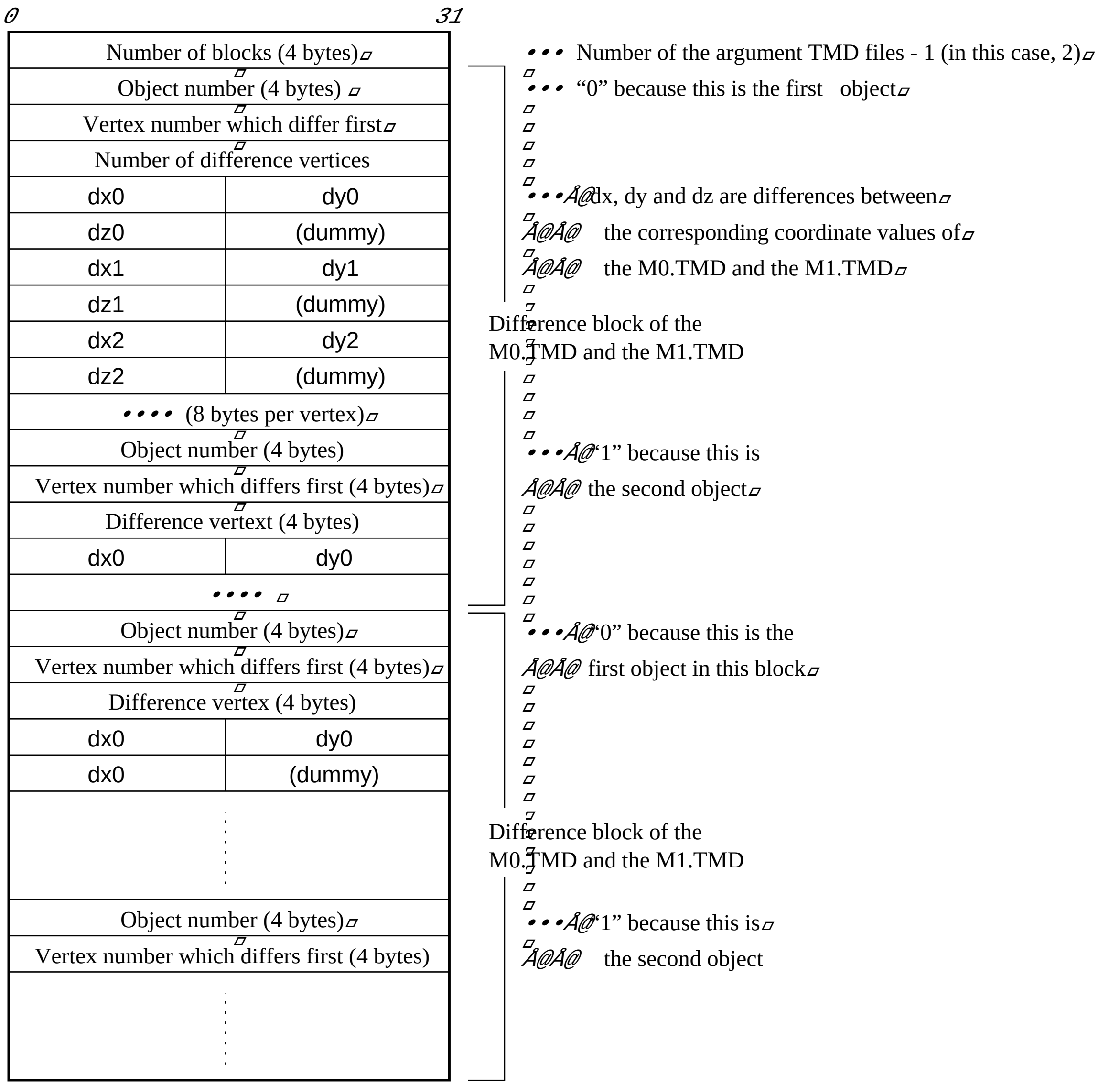


Figure 2-3-1 VDF file format

4 MIMESORT.EXE

1. General

This software reads in multiple TMD files and sorts vertex information. Use of this software before execution of MIMEFILT reduces the difference file. This lets the MIMe animation speed be higher.

2. How to use

MIMESORT [options] BASE.TMD VARIANT1.TMD[VARIANT2.TMD [VARIANT3.TMD...]]

This sorts vertex information in the TMD file and creates a new TMD file in order to reduce a MIMe difference file to be created by MIMEFILET. This allows reduction of calculation time for the MIMe animation run time.

As MIMESORT changes the order of the vertex information, all files to be filtered by MIMEFILT must also be sorted by MIMESORT. Sorting partly only some files with MIMESORT does not produce a correct difference file because vertices between TMD files cannot be corresponded to each other.

The wildcard is available for the argument.

3. Options

-d directory

Specify an output directory in which you want to place the sorted TMD file. The default is "..\STMD". You must create an output directory beforehand.

-v

Outputs the order of vertices before and after sorting to an standard output. This allows you to check how the sorting was. The default outputs nothing. Meanings of outputs are as follows.

file: (vertex: a/b)

file: filename

a : Number of vertices contained in the difference file

b : Total number of vertices

. : Vertices which have no difference and are not contained in the difference file

+ : Vertices which have no difference but are contained in the difference file

* : Vertices which have differences and are contained in the difference file

4. Supplement

- Does not sort any other data than the vertex data (for example, normal data).

- Does not always reduce the difference file. In order to make sure of MIMESORT's effect, filter data both before and after MIMESORT with MIMEFILT to check the size of the difference file. The smaller the size is, the more efficient the data is.

- An input TMD file has to be a file which can be filtered with MIMEFILT. MIMEDORT does not deal with the TMD file which contains multiple objects.

- The wildcard is available for the filename. Specify, however, it so that the base file comes in the head of the file. If a file which has to be a variant file is found in the base file position, an expected sorting result is not acquired. Check whether the wildcard works as it is expected or not with a message "base files=" which is given at execution.

- Creates files just below the output directory even if a sub-directory is specified in the base or variant filename. Therefore, never specify an input file which has the same filename in a different directory.

- Never specify the current directory such as "." to the output directory. If it is specified, an output file overwrites and damages the input file.

- The maximum number of files which can be dealt with is 15. That is, a base file and up to 14 variant files are allowed.

5 MKTOD.EXE

1. General

This software reads in the project information (PRJ file) and hierarchy information (HRC file) to create the layout file (TOD file).

2. How to use

MKTOD [options] hrc-file...

MKTOD is executed in the project directory and reads in the project file (MODEL.PRJ and OBJECT.RPJ) and the HRC file which is given with the argument to create the TOD file in the TOD directory. The TOD file has information about the layout of multiple objects. Unless the TOD directory exists in the current directory, a TOD file is created in the current directory. In both case, both project files must exist in the current file.

The HRC file which is the argument must exist in the current directory (.\) or .\HRC unless a path is specified explicitly. MKTOD searches the HRC file in this order. Omission of the extension ".HRC" is allowed.

3. Options

-o output-file

Specify an output TOD filename. The default creates a filename identical with the argument HRC filename in the current directory except the extension which is replaced with ".TOD".

-tree

Outputs the hierarchical structure of object to a standard output.

-v

Outputs information about what is being converted to a standard output.

4. Constraint

The current version has the following constraints.

- Skips sequence blocks in the HRC file. Therefore, the TOD file which is created by MKTOD is always for a frame (that is, it is not an animation data but layout data.)

5. Note

- Never use the parallel move (-t) or scale-up/down (-s) option of RSDLINK to create a program which uses the TOD file outputted by a animation tool or MKTOD. Specifying the move or scale-up/down option in conversion from the RSD to the TMD causes wrong movement or size of object because the movement and scale factor for the RSD object are stored in the TOD file. Execute the move or scale-up/down with the RSD data with RSDFORM beforehand.

6. Example

Example 1: Display by the -v option

MKTOD is usually started up in a directory (project directory) having sub-directories shown below. The animation tool creates this directory structure and the project file.

```
C:> CD ¥HOME¥PROJ1

C:> DIR /W

 [.]   [..]  [HRC] [ORG] [RSD] [CAM]
 [LHT] [ACT] [TMD] [TIM] [TOD]
 OBJECT.PRJ  MODEL.PRJ

C:> MKTOD -v TEST.HRC

 Input file    : C:¥HOME¥PROJ1¥HRC¥test.hrc
 Output file   : C:¥HOME¥PROJ1¥TOD¥test.tod
 Reading ... C:¥HOME¥PROJ1¥model.prj
 Reading ... C:¥HOME¥PROJ1¥object.prj
 Reading ... C:¥HOME¥PROJ1¥HRC¥test.hrc
 Number of Frames  : 1
 Number of Objects : 21
 Number of Packets : 72

 Output file size  : 1096 bytes
```

As mentioned above, the -v option lets you know output and input filenames, numbers of frames, objects and packets and file size of the created TOD file.

Example 2: Display by the -tree option

The following is a display of the parentage between objects with the -tree option. Names and IDs of all objects written in the hierarchical structure description part of the HRC file are displayed. The child object is indented from the parent object in it.

```
C:> MKTOD -tree HAND.HRC

  = world(#999)
         +- joint1(#1001)
                +- obj1(#101)
                +- joint2(#1002)
                       +- obj2(#102)
         +- joint3(#1003)
                +- obj3(#103)
                +- joint4(#1004)
                       +- obj4(#104)
         +- joint5(#1005)
                +- obj5(#105)
                +- joint6(#1006)
                       +- obj6(#106)
         +- joint7(#1007)
                +- obj7(#107)
                +- joint8(#1008)
                       +- obj8(#108)
         +- joint9(#1009)
                +- obj9(#109)
                +- joint10(#1010)
                       +- obj10(#110)
```

The above is an example when you input the following HRC file (ID is available from the OBJECT.PRJ file.)

```
C:> TYPE HRC¥HAND.HRC
@HRC940401
# Number of objects
       21
# Instance description part
       world     null
       obj1      cube1.rsd
       obj2      cube1.rsd
       obj3      cube1.rsd
       obj4      cube1.rsd
       obj5      cube1.rsd
       obj6      cube1.rsd
       obj7      cube1.rsd
       obj8      cube1.rsd
       obj9      cube1.rsd
       obj10     cube1.rsd
       joint1    null
       joint2    null
       joint3    null
       joint4    null
       joint5    null
       joint6    null
       joint7    null
       joint8    null
       joint9    null
       joint10   null
# Matrix (rotation angle, scale factor, parallel move) initialization part
 world 0
       0 0 0 1.0 1.0 1.0 0 0 8000
 obj1  0
       0 0 0 4.5 4.0 1.0 0 -200 0

       <...intermediate parts are omitted...>

 joint9      0
       0 0 0 1.0 1.0 1.0 -500 -950 0
 joint10     0
       0 0 0 1.0 1.0 1.0 0 -650 0
```

ACT file description part (sequence block)

```
0
```

Hierarchical structure description part

```
world     joint1
joint1    joint2 joint3 joint5 joint7 joint9
joint3    joint4
joint5    joint6
joint7    joint8
joint9    joint10
joint1    obj1
joint2    obj2
joint3    obj3
joint4    obj4
joint5    obj5
joint6    obj6
joint7    obj7
joint8    obj8
joint9    obj9
joint10   obj10
```

6 RSD2DXF.EXE

1. General

This software converts the RSD format file group into the DXF format. This is an MS-DOS Commandline application.

2. How to use

RSD2DXF[options] [ARG-files...]RSD-files...

A filename of the DXF to be outputted is identical with the input RSD filename except the extension which is replaced with ".DXF". The wildcard is available for the argument.

3. Options

-o output-file

Specify an output filename. The default creates a filename identical with the input filename in the current directory except the extension which is replaced with ".DXF". Use of this option is allowed only when only one RSD file is inputted.

-r

Reverses the front and rear of all polygons that are contained in a DXF file to be created.

-h

Shows a list of options which can be specified and an easy-to-use method.

-v

Shows information about conversion.

-ARG-files

Specify an argument file which describes options for RSD2DXF.EXE, input filename and others. With this, the filename having the extension ".ARG" is constructed as the argument file. Use this when the command argument of above 128 bytes is required.

The line starting with "#" is regarded as comment and has no effect on execution of RSD2DXF.EXE.

4. Remark

In a coordinate system of the DXF file to be created, the right side is +X, the upper side is +Y and the rear side is +Z. It is identical with the default which DXF2RSD.EXE and DXFRSDW.EXE suppose as the coordinate system of the DXF file. Therefore, when you read in and edit the DXF file to be created with a modeler, in order to let the model be subject to the coordinate system of the modeler, rotation of the model is required.

Information about the normal, color and texture contained in the RSD file is discarded.

If the same output filename exists, the contents are overwritten. No warning will be given. Pay attention to it.

Even if a argument filename is written in the argument file, it is meaningless.

The wildcard is unavailable in the argument file.

7 RSDCAT.EXE

1. General

This software links multiple RSD file sets to create a new RSD file set. This is an MS-DOS Commandline application.

2. How to use

RSDCAT [options] [ARG-files...] RSD-files...

The wildcard is available for the argument.

3. Options

-o output-file

Specify an output filename. The default creates “OUT.RSD”, “OUT.PLY”, “OUT.MAT” and “OUT.GRP” in the current directory.

Specifying a filename with this option creates outputs identical with the input filename except the extension which is replaced with RSD, PLY, MAT and GRP.

-h

Shows a list of options which can be specified and an easy-to-use method.

-v

Shows information about conversion.

-ARG-files

Specify an argument file which describes options for RSDCAT.EXE, input filename and others. With this, the filename having the extension “.ARG” is constructed as the argument file. Use this when the command argument of above 128 bytes is required.

The line starting with “#” is regarded as comment and has no effect on execution of RSDCAT.EXE.

4. Remark

A group is added to a group file (*.GRP) to be created for each input RSD file. This group consists of all polygons of the input RSD file and has a group name identical with the input RSD filename except the extension which is removed.

RSDCAT.EXE does not compress overlapping textures, vertexes, normals, groups and others through multiple input RSD files. Therefore, files which are outputted by RSDCAT.EXE may require some amendment.

If the same output filename exists, the contents are overwritten. No warning will be given. Pay attention to it.

Even if a argument filename is written in the argument file, it is meaningless.

The wildcard is unavailable in the argument file.

8 RSDFORM.EXE

1. General

This software deforms or moves the 3D model data (RSD) for the artist.

2. How to use

RSDFORM [options] RSD-name

This scales up/down, rotates and moves a RSD format 3D model data to deform the shape of the model. When multiple options are specified at a time, they will be executed in the order of scaling → rotation → parallel move. Omission of the extension ".RSD" of the argument RSD filename is allowed.

3. Options

-o RSD-name

Specify an output RSD filename. The extension is removed. No overwriting to the input file is allowed. The default is "a".

-s x y z

Specify scale factors for x, y and z direction. Specify them with the floating-point numeric value. The value of 1.0 is a scale of 1/1. Specifying the minus number is allowed. In this case, a mirror image (shape at the symmetric position) with respect to the axis is acquired.

-r x y z

Specify rotation angles for the x, y and z axes. Specify them with the floating-point numeric value. The plus value represents the right handed screw direction rotation against the axis (*). The default's angle unit is the degree but once the -rad option is specified, it turns to the radian.

(*) "Right handed screw direction" means a direction to which when you erect the thumb of the right hand and direct its tip to the plus direction of the rotation axis, the remaining fingers wind.

-t x y z

Specify parallel moves for the x, y and z axes. Specify them with the floating numeric value.

-rad

Turns the unit of the rotation angle to the radian.

-l

Fixes the center of the model for scaling and rotation. Specify this together with the -s and/or -r option.

-c

Shares a file, if possible. (Refer to Supplement.)

-c[pmg]

Specify any of PLY, MAT and GRP files to be shared. Combine three characters (p, m and g) to specify them like -cpg (this is equal to -cp -cg). When other options are specified and a file cannot be shared (refer to Supplement), a warning is given and the sharing option is ignored.

-k

Holds the original file version if the input RSD has an old version ID. Unless this option is specified, @ RSD940102, @PLY940102, @ MAT940801 and GRP940102 are given.

-quiet

Adds no comment of deformation history to the output file. The default is OFF.

-v

Outputs detailed information about conversion to an standard output.

4. Supplement

- The RSD data is a set of four files, .RSD, .PLY, .MAT and .GRP. These files must exist in the same directory.

- Each file can be shared when the following condition is met.

  PLY file: No deformation is specified.
  MAT file: No mirror image against the original data
  GRP file: Can always be shared.

  Provided that all files to be shared must exist in the same directory.

- When you create a program which uses the TOD file, use this tool but RSDLINK for moving or scaling up/down. (Refer to Chapter2, Section 9: RSDLINK.EXE)

- The PlayStation coordinate system uses the sign 16-bit. This rule does not apply to the RSD but you must convert it into the TMD with RSKLINK so that it can settle in the coordinate system.

5. Example

Example 1: Basic use

Scale up it in the x axis direction by 10 times and then rotate it around the x axis by 45 degrees. Specify the file sharing if possible.

```
C:> RSDFORM -v -s 10 1 1 -r 45 0 0 -c CUBE
```

Output	Description
`Input  RSD   : CUBE` `Output RSD   : a`	

`Scale        : 10,    1,    1`
`Rotation     : 45,    0,    0 (degree)`
`Translation  :  0,    0,    0`
	Contents of deformation

`Range      x :   +0.0 ... Â@+1000.0 (center: Â@+500.0)Â@`
`           y : -500.0 ... Â@ +500.0 (        Â@  +0.0)Â@`
`           z : -500.0 ...  Â@+500.0 (        Â@  +0.0)Â@`
	Size before deformation (Maximum and minimum values of each axis and co-ordinate values of the center)

`Â@Â@CUBE.ply ->       Â@ a.ply  Â@Â@Â@Â@Â@Â@Â@Â@Â@Â@Â@Â@`	Output filename
`Â@Â@CUBE.mat ->  Â@Â@CUBE.mat(Shared)Â@Â@Â@Â@Â@Â@Â@Â@Â@Â@Â@Â@Â@`	MAT file can be shared
`Â@Â@CUBE.grp ->  Â@Â@CUBE.grp(Shared) Â@Â@`	GRP file can be shared
`Â@Â@CUBE.rsd ->        Â@a.rsd`	

`New Range  x :   +0.0 ... +10000.0 (center: +5000.0)`
`           y : -707.1 ...   +707.1 (           +0.0)Â@`
`           z : -707.1 ...   +707.1 (           +0.0)Â@Â@`
	Size after deformation (Maximum and minimum values of each axis and co-ordinate values of the center)

Example 2: To deform partly

Apply the -l option in the example 1 deformation. With this option, you can specify the center of gravity of the model as the center of rotation and scaling. Location of the model does not change. Compare this with the result of the example 1.

```
Input  RSD  : CUBE
Output RSD  : a

Scale       :    10,    1,    1
Rotation    :    45,    0,    0 (degree)
Translation :     0,    0,    0
Range     x :      +0.0 ... +1000.0 (center: +500.0)
          y :    -500.0 ...  +500.0 (          +0.0)
          z :    -500.0 ...  +500.0 (          +0.0)

   CUBE.ply ->           a.ply
   CUBE.mat ->           a.mat
   CUBE.grp ->           a.grp
   CUBE.rsd ->           a.rsd

New Range x :   -4500.0 ... +5500.0 (center: +500.0)
          y :    -707.1 ...  +707.1 (          +0.0)
          z :    -707.1 ...  +707.1 (          +0.0)
```

The center of gravity is

```
C:> RSDFORM -v -l -s 10 1 1 -r 45 0 0 CUBE
```

Example 3: Copying the RSD file

This software can be used as a copying tool for the RSD file. This is also available for change of the RSD name. It is useful for you to create the following batch file.

```
C:> TYPE RSDCOPY.BAT
    @ECHO OFF
    RSDFORM -o %2 %1
```

Example 4: Overwriting of the RSD file

This tool protects the pre-deformation RSD file from being overwritten. However, you can use this tool as an overwriting tool with the following batch file.

```
C:> TYPE RSDFORM2.BAT
    @ECHO OFF
    SET ARGS=
    REMÅ@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@   Read in all arguments.
     :LOOP1
    IF "%9"=="" GOTO LABEL1
    SET ARGS=%ARGS% %1
    SHIFT
    GOTO LOOP1
     :LABEL1
    REMÅ@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@   Convert it with a temporary na
    RSDFORM -o _TMP %ARGS% %1 %2 %3 %4 %5 %6 %7 %8
    IF ERRORLEVEL 1 GOTO END
    REMÅ@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@   Search the argument (= input f
    :LOOP2
    IF "%1"=="" GOTO LABEL2
    SHIFT
    GOTO LOOP2
     :LABEL2
    REMÅ@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@   Overwrite the input file (%0)
    RSDFORM -o %0 -quiet _TMP
    REMÅ@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@Å@ Delete the _TMP.* to finish it.
    DEL _TMP.*
    :END
    SET ARGS=
```

9 RSDLINK.EXE

1. General

This software converts the 3D model data (RSD) for the artist into the 3D model format (TMD) which the libgs can deal with.

2. How to use

RSDLINK [options] rsd-names...
RSDLINK [options] rsd-names [options] rsd-names [options]...
RSDLINK [options] arg-files...
RSDLINK [options] prj-file

These link multiple RSD data you specify with the argument to create a TMD file. This allows you to set a scale factor and parallel move for each RSD data. (Refer to example 1.)

Unless an argument RSD name contains a path, first the current directory (.\) and then the .\RSD directory are searched.

If there are many arguments, you can write them in a file (*.ARG) to give them. Omission of ".RSD" is allowed but ".ARG" of the argument filename must not be omitted. (Refer to example 2.)

Similarly, you can give directly a project file (*.PRJ). This case is equal to a case when you give all of the RSDs contained in the project file with the argument. At the same time, the -id option is regarded to be specified. Omission of ".PRJ" is not allowed.

3. Options

-o file-name

Specify an output filename. The default is "a.tmd".

-s factor

Scales up or down the RSD data after the next argument (*).

-sc factor

The scale factor is rounded to powers of 2. The default is 1.0. Set a proper scale factor so that it can settle in the TMD format coordinate system because the coordinate values in the TMD format are 16-bit integers.

-t x y x

Moves the RSD data after the next argument (*). The default is (0 0 0).

-id PRJ-file

Specifying the project file (MODEL.PRJ) creates an array of object ID numbers for each object to be created in the TMD file. The result is created in the current directory as a header file to the C language. The output filename is identical with the TMD filename except the extension which is replaced with ".h" (refer to example 3.)

-info

Outputs very detailed information about the kind, vertex coordinate values and textures of objects which are being converted to an standard output (refer to example 5.)

-v

Outputs detailed information about conversion such as the number of polygons to a standard output. Moreover, as their approximate sizes (Range: minimum and maximum values of the vertices (x, y and z)) in the PlayStation coordinate system are also outputted, you can check their positions or sizes before displaying them on the PlayStation board (refer to example 4.)

(*) Note: The -s and -t are effective to all the RSDs which appear after them unless they are specified again.

4. Constraint

The current version has the following constraints.

- May not link the RSD data if there are too many polygons in a RSD data. The upper limit of the number of polygons depends on the numbers of vertices and normals and the free memory capacity but it is usually about 5000 polygons. However, this limit applies only to "a RSD" and there is, in practice, no limit to the number of the RSDs to be linked and number of polygons of the whole TMD to be created.

- If a RSD which does not appear in the project file is linked, "-1" is allocated to the header file.

- If you specify the same RSD twice when you specify a project file, the ID value is allocated to only the first RSD and "-1" is allocated to the second and following RSDs.

- A line of the RSD file must consist of 1024 bytes or less.

CHAPTER 2

5. Note

- Never use the parallel move (-t) or scale-up/down (-s) option to create a program which uses the TOD file outputted by an animation tool or MKTOD. Specifying the move or scale-up/down option in conversion from the RSD to the TMD causes wrong movement or size of object because the movement and scale factor for the RSD object are stored in the TOD file.
- For the move and scale-up/down, use DXF2RSD (-sc -t) or RSDFORM (-s -t) in the RSD format, unless otherwise specified. You can acquire the more correct and easy-to-use data.

6. Supplement

- The image data file (TIM file) which is described in the RSD file must exist in the same directory (.\) as the RSD file exists or ..\TIM directory. RSDLINK searches the TIM file in this order.
- The following table shows what TMD type the RSD is converted to by RSDLINK. Heads and items show types which are described in the MAT file. (Tri, Quad, Line, Sprite and others are specified with numerals in the PLY file.) The 0x... is a header of the TMD packet. In the table, only one-sided polygons which are not semitransparent are shown. Packet of the semitransparent or both sided polygons is the same though the header has a different value. For detail, refer to “Appendix: File format RSD and RMD”.

		åıåπåvéZÇ†ÇË		åıåπåvéZÇ»Çµ	
	Shading Å®	Flat	Gouraud	Flat	Gouraud
É}ÉeÉäÉAÉãÅ{É/ÉäÉSÉìÅEÉ^ÉCÉv	C+Tri	0x20000304	0x30000406	0x21010304	(0x21010304)
	G+Tri	0x20040506	0x30040606	(0x31010506)	0x31010506
	T+Tri	0x24000507	0x34000609	(0x25010607)	(0x25010607)
	D+Tri	(0x24000507)	(0x34000609)	0x25010607	(0x25010607)
	H+Tri	(0x24000507)	(0x34000609)	(0x35010809)	0x35010809
	C+Quad	0x28000405	0x38000508	0x29010305	(0x29010305)
	G+Quad	0x28040708	0x38040808	(0x39010608)	0x39010608
	T+Quad	0x2c000709	0x3c00080c	(0x2d010709)	(0x2d010709)
	D+Quad	(0x2c000709)	(0x3c00080c)	0x2d010709	(0x2d010709)
	H+Quad	(0x2c000709)	(0x3c00080c)	(0x3d010a0c)	0x3d010a0c
	C+Line	(0x40010203)	(0x40010203)	0x40010203	(0x40010203)
	G+Line	(0x50010304)	(0x50010304)	(0x50010304)	0x50010304
	T+Sprite	(0x64010305)	(0x64010305)	0x64010305	(0x64010305)
	1x1	(0x6c010204)	(0x6c010204)	0x6c010204	(0x6c010204)
	8x8	(0x74010204)	(0x74010204)	0x74010204	(0x74010204)
	16x16	(0x7c010204)	(0x7c010204)	0x7c010204	(0x7c010204)

Table 2-9-1 RSD → TMD conversion comparison table

TMDs mentioned in the parentheses are for improper combinations of material types. Some options are ignored to change them to proper TMDs.

7. Example

Example 1: use

```
C:> RSDLINK -v -o BOXES.TMD BOX1 -s 2.0 -t 100 100 100
        BOX1 BOX2 -t 200 -200 200 BOX3
```

In this case, four objects shown below are put together into a TMD file (BOXES.TMD).

BOX1: As BOX1 is
BOX1': BOX1 is doubled (100 100 100) and moved
BOX2: BOX2 is doubled (100 100 100) and moved
BOX3: BOX3 is doubled (200 200 200) and moved

Example 2: How to put together arguments in a file

You can also put together arguments in a file as follows.

```
C:> TYPE TEST.ARG
       BOX1
       -s 2.0 -t 100 100 100 BOX1 BOX2
       -t 200 -200 200 BOX3
C:> RSDLINK -v -o BOXES.TMD TEST.ARG
```

Example 3: Read-in of project file and creation of ID array

Project file ("MODEL.PRJ") is an ID table of objects which an animation tool creates.

```
C:> TYPE MODEL.PRJ

@PRJ940701
# Next available ID
 6

# ID   Object Name
       0     head.rsd
       1     body.rsd
       2     l_arm.rsd
       3     r_arm.rsd
       4     l_leg.rsd
       5     r_leg.rsd
```

If there is a file mentioned above and you execute

```
C:> RSDLINK -o ROBO.TMD MODEL.PRJ
```

or

```
C:> RSDLINK -o ROBO.TMD -id MODEL.PRJ BODY HEAD  L_ARM R_ARM
L_LEG R_LEG
```

ROBO.H having the following contents is created together with ROBO.TMD. IDs of objects are registered as elements of the array in the same order they appear in the TMD. You can use this when you want to access each object in the TMD from the application program.

```
C:> TYPE ROBO.H
      /*
       *   TMD ID List for "ROBO.TMD"
       *
       *   Created by rsdlink(v3.6) at 10:10 09/16/94
       *
       *   (C)1994 Sony Computer Entertainment Inc. All Rights
Reserved.
       */
       int robo_list[] = {
               1,    /* body  */
               0,    /* head  */
               2,    /* l_arm */
               3,    /* r_arm */
               4,    /* l_leg */
               5,    /* r_leg */
       };
```

Example 4: Outputs when the -v option is used

```
C:> RSDLINK -v DINO -s 100 BOX
```

Output	Description
`[0] ============== RSD ==============`	The first RSD (“DINO.rsd”)
`  RSD files   : \PSXGRAPH\DATA\RSD\DINO.ply, DINO.mat, DINO.grp`	
`       TEX[0] = dino0.tim`	
`       TEX[1] = dino1.tim`	
`       TEX[2] = dino2.tim`	Texture filename
`       TEX[3] = dino3.tim`	
`       TEX[4] = dino4.tim`	
`       TEX[5] = dino5.tim`	
`  POLYGON     : 2724`	Number of polygons
`  VERTEX      : 1376`	Number of vertices
`  NORMAL      : 2671`	Number of normals
`  MATERIAL    : 2592`	Number of materials
`  Range       : (-180, -210, -1690)-(180, 580, 290)`	Minimum and maximum values of a range (x, y and z)
`[1] ============== RSD ==============`	
`  RSD files   : \PSXGRAPH\DATA\RSD\BOX.ply, BOX.mat, BOX.grp`	The second RSD (“BOX.rsd”)
`  POLYGON     :   12`	
`  VERTEX      :    8`	
`  NORMAL      :   12`	
`  MSTERIAL    :    1`	
`  Scale       :  128`	Scale factor (“100” is rounded to the n-th power of 2)
`  Range       : (-6400, -6400, -6400)-(6400, 6400, 6400)`	
`================= TMD ==================`	
`  Output TMD  : "a.tmd"`	Output filename
`  TMD header  :       (    12 bytes)`	TMD file header size
`  Objects     :      2(    56 bytes)`	Total number of RSDs
`  Primitives  :   2736( 65640 bytes)`	Total number of primitives
`                    12  ...Flat Colored Triangles`	
`                   136  ...Gouraud Colored Triangles`	Breakdown by mode
`                  2434  ...Flat Textured Triangles`	
`                   154  ...Gouraud Textured Triangles`	
`  Verteices    :  1384 ( 11072 bytes)`	Total number of vertices
`  Normals      :  2683 ( 21464 bytes)`	Total number of normals
`--------------------------------------`	
`Total file size:         98244 bytes`	Output file size

Example 5: Outputs when the -info option is used

You can check what TMD the data is actually converted to with the -info option.

```
C:> RSDLINK -info BOX
```

Output	Description
`Input RSDs       : 1 object(s)`	Number of objects in the TMD
`        RSD[ 0] = "BOX"`	Each RSD name
`Total vertices   :    8`	Total number of vertices
`Total normals    :   12`	Total number of normals
`Total primitives :   12`	Total number of primitives

```
---------------------< BOX >---------------------
= [  0] FLAT TEX 3-POLY(0x24000507) LIGHT: ON =
        Vert-0: (  -150,   -150,   -150) (#2)
        Vert-1: (   150,   -150,   -150) (#6)
        Vert-2: (  -150,    150,   -150) (#0)
        Norm-0: (     0,      0,  -4096) (#0)
        UV 0-2: (  0   0) ( 47   0) (  0  47)
        Pixel mode  : 4bit CLUT : (x y)=(  0 480)
        Texture Page: 10  Texture No. :  0
= [  1] FLAT TEX 3-POLY(0x24000507) LIGHT: ON =
                ÅEÅEÅE
```

For each primitive,

```
= [0] FLAT TEX 3-POLY(0x24000507) LIGHT: ON =
```

that is, the [number of polygons], FLAT/GOURAUD, TEX/NO TEX, SEMITRANSPARENCY ON/OFF, TWO-SIDED/ONE-SIDED, GRADATION, types of primitive (3-POLY, 4-POLY, SPRITE), hexadecimal display of primitive header (0x...) and LIGHT calculation ON/OFF) are displayed.

Then, coordinate values of each vertex which forms the primitive are displayed as shown below.

```
Vert-0: (-150, -150, -150) (#2)
```

(#...) is a vertex number used in the PLY file.

Similarly, the normal is displayed.

```
Norm-o: (0, 0, -4096) (#0)
```

The RSD normal is usually normalized to size 1 (in this case, (0, 0, -1.0)) but as the floating point in the TMD has a 4096 value for 1 of the RSD, it is displayed as mentioned above.

Then, material information such as UV coordinates and color of the texture is displayed.

10 TMD2PMD.EXE

1. General

Input of the TMD format file (*.TMD) which is the PlayStation 3D modeling data format creates the PMD file (*.PMD) which is the high speed modeling data format. For detail of PMD format, refer to “Appendix: File format: PMD”. This software is an MS-DOS Commandline application.

2. How to use

TMD2PMD [options] TMD-files...

The wildcard is available for the argument. You can input and convert multiple files at a time. In this case, for each inputted filename, a filename is created in the current directory which is identical with it except the extension which is replaced with “.PMD”.

3. Options

-o output-file

Specify an output filename. The default creates a filename identical with the input filename in the current directory, except the extension which is replaced with “.PMD”

Input of two or more files does not allow use of this option. If you specify the wildcard which applies to two or more files, use of this option causes an error.

-h

Shows a list of options which can be specified and an easy-to-use method.

-v

Shows detailed information about conversion.

-s

Sets vertex data in the PMD file to be created in sharing vertex mode.

+s

Sets vertex data in the PMD file to be created in independent vertex mode. (Default)

-l

Turns ON the light calculation flag in the PMD file to be created.

+l

Turns OFF the light calculation flag in the PMD file to be created. (Default)

-b

Turns ON the Back clip flag in the PMD file to be created. (Default)

+b

Turns OFF the Back clip flag in the PMD file to be created.

-notge

Turns OFF the TGE flag of all packets in the PMD file to be created.

+notge

For packet TGE flag in the PMD file to be created, those in the input TMD file are copied. (Default)

4. Remark

Texture polygon of Light calculation ON in the TMD file does not contain information about the color. TMD2PMD.EXE regards that all of R, G and B in such data have the brightness of 0x80. and creates the PMD file.

11 TMDINFO.EXE

1. General

This software shows contents of the 3D model data TMD file.

2. Hot to use

TMDINFO [-vl-l] TMD-file

This outputs numbers of objects, polygons (breakdown by packet type), vertices and normals in the TMD file you specified with the argument to a standard output. This is useful to check contents of the TMD file. Omission of ".TMD" is allowed.

3. Options

-v or -l

Outputs very detailed information about the type, vertex coordinate values and texture of all packets to a standard output. The display contents depend on the output of "Chapter 2, Section 9: RSDLINK.EXE". Refer to RSDLINK.TXT.

4. Constraint

- The number of vertices (and normals) per object must be 8189 or less when the -v option is used. Unless the -v option is used, this constraint is meaningless.

12 TMDSORT.EXE

1. General

This software sorts the 3D model data file (TMD) by packet type. This minimizes branching by packet type at painting. This is used to realize a high painting speed.

2. How to use

TMDSORT [options] TMD-file

This sorts packets in each object in the TID file you specified with the argument and outputs the result into a new TMD file. Continuation of packets of the same type minimizes branching to realize a high painting speed. Omission of ".TMD" is allowed.

3. Options

-o file-name

Specify an output TMD filename. The default is "a.tmd". However, overwriting to the input file is not allowed.

-n

Executes no sorting but outputs information about the input TMD to a standard output.

-v or -l

Outputs very detailed information about the type, vertex coordinate values and texture of all packets to a standard output. The display contents depend on the output of RSDLINK-info. Refer to "Chapter 2, Section 9: RSDLINK.EXE".

4. Constraint

- The number of vertices (and normals) per object must be 8189 or less when the -v option is used. Unless the -v option is used, this constraint is meaningless.

- Sorting is executed, object by object in the TMD.
  A different object is not sorted even if it exists in the same TMD.

CHAPTER 3

MATERIAL EDITOR (MEDITOR.EXE)

This chapter explains all the functions supported by the material editor.

1 Outline

The material editor is a tool used to edit material for three-dimensional models.
The material means the following surface attributes that can be set for each polygon.

- Color
- Texture
- Transparency
- Shading method (Flat/smooth)
- Availability of light source calculation

Note:

For installing the material editor, see part 1, Installation.

Note:

The material editor does not support a function for creating texture.
Applying texture requires that the texture have been created using a sprite editor, etc.

Note:

This manual covers material editor version 1.7. An older version should be replaced by the latest one.

2 Operating Environment

As operating a three-dimensional model displayed on the video monitor, the user edits material. The artist board is used to render a three-dimensional model. This requires that the artist board have been installed in the host computer to be used.

The keyboard for the host computer is used to operate models. Materials are to be selected on Windows.

3 Fundamental Use

The following gives the basic method of operating the material editor.

1) Start up the material editor.

2) Select “Open” in the file menu. The dialogue box used to select a file is displayed.

3) Select the file to be read in, and click the OK button.
   With the texture model set, the texture arrangement dialogue boxes corresponding to the number of textures used are displayed. Enter an appropriate value into the dialogue box.

4) If needed, operate the three-dimensional model displayed on the video monitor for movement to the desired location and direction. The keyboard and the spin button on the moving dialogue box can be used to move the model.

5) Select the material dialogue box for the material menu.

6) Click the polygon selection button. The pointer is moved onto the video monitor, with the model for polygon selection set up.

7) Select the polygon for setting the material.

8) Click the right button on the mouse to return the pointer onto Windows.

9) Click the radio button in the dialogue box to set the material.

10) Repeat above steps 6) to 9) to set materials for desired polygons.

11) Select “Storage” or “Storage under another name” to save the file.

4 File Format

RSD format

The material editor loads/saves 3D model data described in the RSD format. The RSD format is the standard format for the PlayStation 3D graphics tool group, allowing the description of 3D model shapes and materials. The RSD format is such that four kinds of files are used to describe a model. Specifications of the RSD format are covered in the appendix of this manual.

RSD file	Describes relationships between the PLY/MAT/GRP file and the texture file.
PLY file	Describes the shape of a model.
MAT file	Describes material information of a polygon.
GRP file	Describes grouping information of a polygon.

Table 3-4-1 RSD File

The use of the material editor requires that the RSD file have been created. The 3D graphics tool includes a converter for converting the DXF format (format, for describing 3D shapes, supported by many commercially available modelers) into the RSD format. Thus, the DXF file created by a commercially available modeler can be used for 3D model data. For details, see 3D model creation in chapter for tutorial.

trueSpace for PlayStation (DTL-S280) supplied by Caligari Inc. supports the RSD format. Thus, the RSD file can be handled directly.

TIM format

The material editor uses 2D image data described in the TIM format for texture mapping. The TIM format can be created/edited by our sprite editor. Converters for the BMP, PICT and RGB formats are included in the 3D graphics tool.

5 Directory Structure

RSD and TIM data to be used by the material editor must be stored into the directory in accordance with the following rules.

- Only a file name must be described for PLY/MAT/GRP in the *.RSD file. No absolute path can be used.

Correct example 1)

```
@RSD940102
PLY=HELI.PLY
MAT=HELI.MAT
GRP=HELI.GRP
```

Correct example 2)

```
@RSD940102
PLY=HELI01.PLY
MAT=HELI02.MAT
GRP=HELI03.GRP
```

* File names for RSD/PLY/MAT/GRP can be different.

Incorrect example

```
@RSD940102
PLY=C:¥TEST01¥RSD¥HELI.PLY
MAT=C:¥TEST01¥RSD¥HELI.MAT
GRP=C:¥TEST01¥RSD¥HELI.GRP
```

* The absolute path is invalid.

- The PLY/MAT/GRP files referenced in the *.RSD file must exist in the same path and their directory name must be RSD.

Correct example

D:¥GAME1¥TEST1¥HELI01.RSD (See HELI01.PLY, HELI01.MAT, HELI01.MAT.)
D:¥GAME1¥TEST1¥RSD¥HELI01.PLY
D:¥GAME1¥TEST1¥RASD¥HELKI01.MAT
D:¥GAME1¥TEST1¥RSD¥HELI01.GRP

Incorrect example 1)

D:¥GAME1¥TEST1¥HELI01.RSD (See HELI01.PLY, HELI01.MAT, HELI01.MAT.)
D:¥GAME1¥TEST1¥HELI01.PLY
D:¥GAME1¥TEST1¥HELI01.MAT
D:¥GAME1¥TEST1¥HELI01.GRP

* The parent directory name is not RSD.

Incorrect example 2)

D:¥GAME1¥TEST1¥RSD¥HELI01.RSD
D:¥GAME1¥TEST1¥RSD¥HELI01.MAT
D:¥GAME1¥TEST1¥RSD¥HELI01.GRP

* No PLY file exists.

- The texture file must be associated brotherly with the RSD data storage directory having a name of TIM.

Correct example

D:¥GAME1¥TEST1¥RSD¥HELI01.RSD (See BODY1.TIM, BODY2.TIM.)

…

D:¥GAME1¥TEST1¥TIM¥BODY1.TIM
D:¥GAME1¥TEST1¥TIM¥BODY2.TIM

Incorrect example

D:¥GAME1¥TEST1¥RSD¥HELI01.RSD (See BODY1.TIM, BODY2.TIM.)

…

D:¥GAME1¥TEST1¥RSD¥BODT1.TIM
D:¥GAME1¥TEST1¥RSD¥BODT2.TIM

* The TIM file does not exist in the TIM director associated brotherly with the RSD directory.

Following the above three rules, store the RSD and TIM file groups in advance. Creating a directory for each game tile and scene to store related RSD and TIM data under the directory will allow data to be processed readily.

Example

C:¥GAME01¥SCN00¥RSD
C:¥GAME01¥SCN00¥TIM
C:¥GAME01¥SCN01¥RSD
C:¥GAME01¥SCN01¥TIM
C:¥GAME02¥SCN00¥RSD
C:¥GAME02¥SCN00¥TIM

…

6 Location of Texture Data on VRAM

Texture mapping under the PlaySTation architecture requires that texture data have been located on the VRAM.

How various image data should be located on the limited-size VRAM affects the design of a whole game program. Thus, in the initial phase of authoring, it is difficult to completely fix texture location,

The RSD format provides only for the use of a TIM file as texture data. (VRAM address information is not covered in the RSD file.) The address is always relocatable. Modifying the VRAM address of applied texture never affects RSD data. For locating texture data, see part 2, chapter 3, Creation of Texture Data.

7 Ports and Addresses on Artist Board

The material editor uses a value written in the following file as the port address of the artist board.

C:¥WINDOWS¥ABOARD.INI

For example, if

addr=0x1340

is written in this file, the material editor uses a port address of 0x1340 to communicate with the artist board. If the port address of an actual board is not set at 0x1340 by the dip switch, the material editor cannot communicate correctly with the artist board.

If not set at 0x1340, use the ABOARD.EXE tool to modify file ABOARD.INI, or change the address of the board to set up the same address value.

For the ABOARD.EXE tool, see part 3, reference 5.1, ABOARD.EXE. For changing the port address of the artist board, refer to the manual for the artist board.

8 File Menu

Open

Reads in a new 3D model.

[Operation]

Select “Open”.

In the displayed dialogue box, select the RSD file to be invoked, and click the OK button.

On the material editor, the model is adjusted automatically for an appropriate size for easy operation.

With a texture applied on the invoked 3D model, the dialogue box for setting the address of texture data is displayed. (Thus, data can be located on the VRAM.)

Dialogue box for texture location

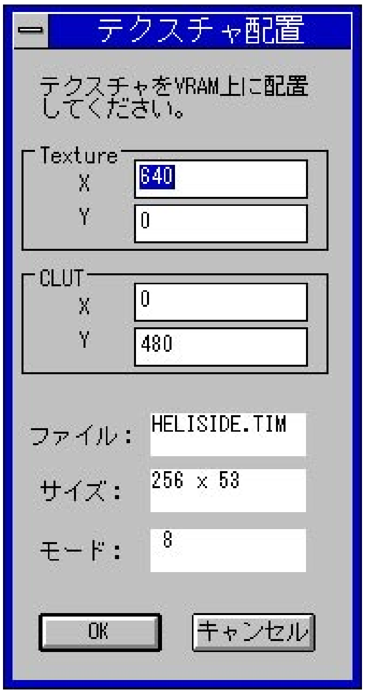


Figure 3-8-1 Dialogue Box for Texture Location

- **Texture (X, Y):**

  Enter X and Y coordinates on the VRAM where image data is to be located. The initial value is X and Y coordinates of the image section set in the TIM file.

- **CLUT (X, Y):**

  Enter X and Y coordinates on the VRAM where the pallet is to be located. The initial value is X and Y coordinates of the pallet section set in the TIM file.

- **File:**

  Name of the invoked TIM file

- **Size**

  Lateral and longitudinal pixel sizes of the image data section

- **Mode**

  Indicates the number of colors.

  4 (bits): 16 colors
  8 (bits): 256 colors
  16 (bits): 32768 colors

- **OK button**

  Loads a texture to the VRAM address specified in the dialogue box for texture location.

- **Cancel button**

  Cancels the loaded texture.
  Loaded RSD data is also canceled.
  As the VRAM address of texture data is relocatable, the set value is merely a tentative address.
  The material editor uses the top left area of the VRAM for rendering and display. Locating texture data on the area for rendering and display results in the destruction of the data.
  The area for rendering and display depends on the screen resolution.

Screen resolution Lateral x longitudinal	Area for rendering and display Top left - Bottom right
256 x 240	(0, 0) - (244, 479)
320 x 240	(0, 0) - (319, 479)
512 x 240	(0, 0) - (511, 479)
640 x 240	(0, 0) - (639, 479)

For resolution of 640 x 240, locating image data in the VRAM address (640, 0) causes no problem. But locating image data in (639, 0) results in the destruction of one left edge line of image data.

For know-how on other texture location, see part 2, chapter 3, Creation of Texture Data.

Reading in vertex

With the RSD file read in, reads in only vertex information from another model to modify only the model shape.

[Operation]

Select “reading in vertex”.

In the displayed dialogue box, select the PLY file to be invoked, and click the OK button.

[Application]

If there are two or more models arranged so as not to damage the number of vertices and the status of connection, material is set for only one of the models.

Reading in vertices from the other models to be saved under other names allows the application of the same material to more than one derivative model. The requirement is that the material be set only once.

Saving

Overwrites opened RSD data.

Saving under another name

Saves currently opened RSD data under another name. An existing file with the name is overwritten. If there is mo file with the name, a new file is created.

[Operation]

Select “saving under another name”. In the dialogue box, enter a file name and clock the OK button.

[Note]

Saving texture-mapped RSD data into another RSD directory makes it impossible to look into the TIM file. Therefore, copy the TIM file referenced into to the TIM directory.

TIM reload

Reloads the VRAM with all the TIM files currently used as textures for the loaded model.

Use this function if another tool that uses the artist board has destroyed texture data on the VRAM.

Snap shot

Saves the image currently displayed on the video monitor into a file in the TIM format.

The image file can be edited by the sprite editor for use as texture. This function enables texture data to be created.

[Operation]

Select “Snap shot”.
In the dialogue box, enter a TIM file name.

End

Terminates the material editor.

9 Material Menu

The material dialogue box pops up. The dialogue box is used to set all materials.

Figure 3-9-1 Material Setting Dialogue Box

RSD file: Name of file serving model being edited
Number of polygons: Total number of polygons in the model being edited
Polygon ID: ID of the last selected polygon

[Operation]

Select a polygon and use the radio button in the material dialogue box to specify attributes of the polygon.

Shading

Sets a polygon shading method.

“Flat”: Flat shading. Clears a polygon boundary.

“Smooth”: Smooth shading. Smoothes a polygon boundary.

In the following Figure, the left model is subjected to flat shading, while the left model is subjected to smooth shading.

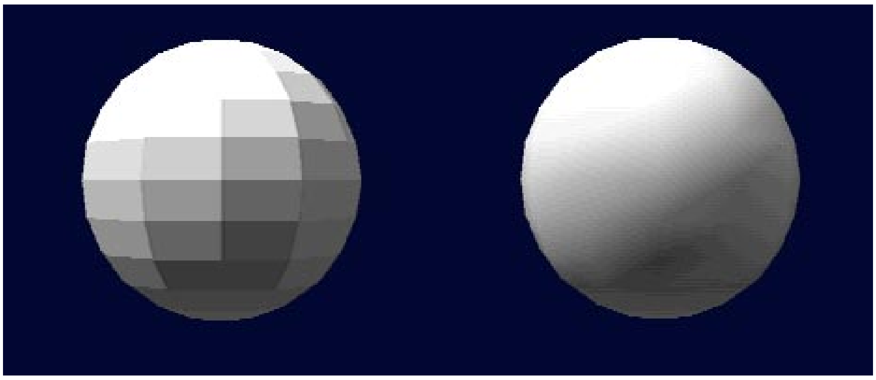


Figure 3-9-2 Shading effects

Note

Smooth shading can be applied more effectively to polygons which are adjacent to each other with a larger angle. To the contrary, a smaller angle results in a darker edge, See the left model in Figure 3.9.3.

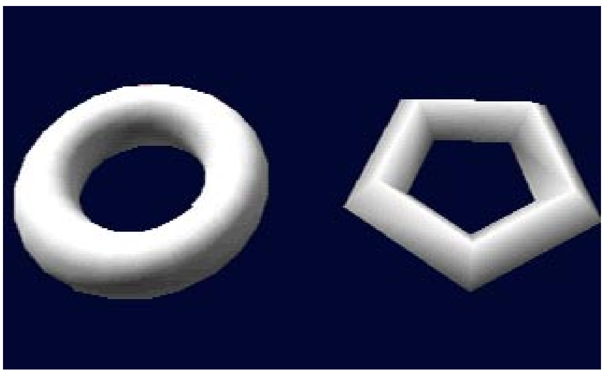


Figure 3-9-3 Angles between Polygons and Smooth Shading

Semitransparency

Sets the degrees of semitransparency for a polygon.

“Semitransparent”: Makes a polygon semitransparent.

“Opaque”: Makes a polygon opaque.

In Figure 3.9.4, the left cube is displayed semitransparently, while the right polygon is displayed opaquely.

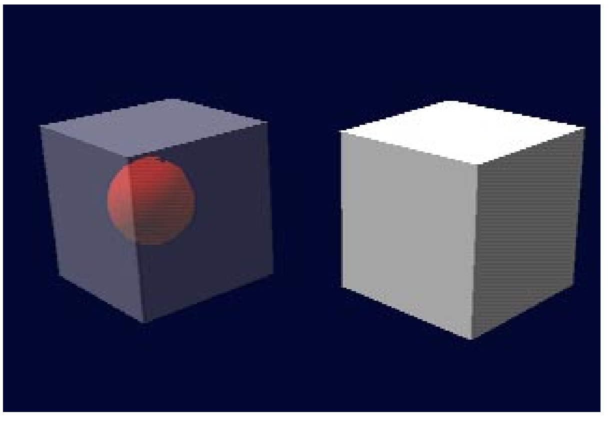


Figure 3-9-4 Semitransparency

Note

Making a texture polygon (provided with applied texture) semitransparent requires that the TIM utility have been used to set the texture semitransparency control (STP) bit at 1.

Texture mapping

Sets the color or texture map of a polygon.

"Texture": Applies texture. The procedure of texture mapping is explained in detail in the next section.

"Color": Applies color. The dialogue box for setting color pops up. Select the color to be applied. In the polygon selection mode, filling and brushing functions can be used.

Filling function: Shift + Ctrl + right button click
Stores the polygon color as the current brush color.

Brushing function: Shift + right button click
Applies the brush color to a polygon.

Light source calculation

Specifies whether to carry out light source calculation.

"On": Carries out light source calculation.

"Off": Does not carry out light source calculation.

For example, light source calculation provides three-dimensional appearance. Without light source calculation, the same color is applied to the whole polygon, providing no three-dimensional appearance.

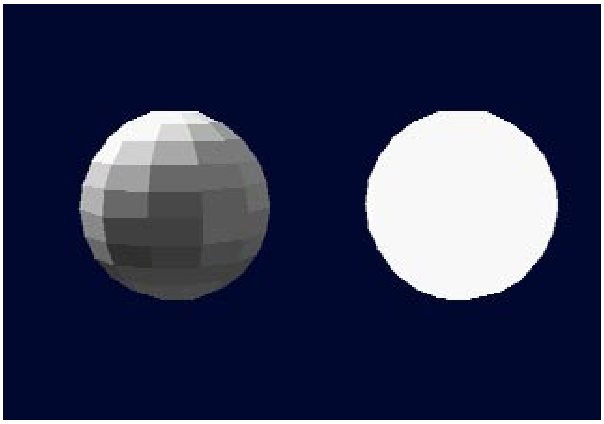


Figure 3-9-5 Light Source Calculation

Application

With a partial model polygon colored brightly, no light source calculation is applied only the polygon. Damping light for the model brightens only the part subjected to no light source calculation. The part provides lighting effects.

Display

Specifies whether to display the selected polygon.

“On”: Displays the polygon.

“Off”: Does not display the polygon.

The display of an off polygon is suppressed until the specification of on. Further, such a polygon cannot be selected by clicking the mouse.

Selecting a polygon whose display is suppressed requires the selection of all polygons by the F3 key or of a cataloged group. Releasing display suppression allows the polygon to be displayed.

This function can be used validly to mask the polygons other than the target polygon when it is difficult to select the target polygon because of the other obstacle polygons or when the target polygon is covered by the other polygons.

“Copy” button

Copies the selected polygon.

A new created polygon shares the vertices with the old polygon.

Both the polygons have the same normal and material attributes. But these values can be changed independent of one another.

The copying function is applicable to sealing and the creation of double-faced polygons.

a) Sealing function

The surface of a plastic model can be sealed with texture.

Applying texture with a transparent part onto the upper polygon combined with the lower polygon leads to the observation of the lower polygon through the transparent texture part. The effective use of the sealing function allows the significant reduction of the number of textures on the VRAM.

In the following example, two kinds of seals are applied onto the same-quality texture.

Figure 3-9-6 Example 1) of Sealing Function

The sealing function can bridge over difficulties of differently colored texture polygons. In the following example, the left polygon is provided with typical texture, while the right polygon is provided with a seal.

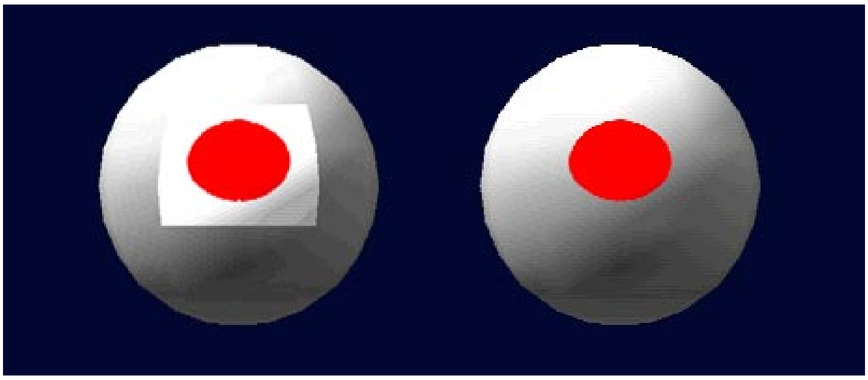


Figure 3-9-7 Example 2) of Sealing Function

b) Creation of double-faced polygon

Reversing a copied polygon enables the creation of a double-faced polygon. The following reversing button is used to reverse a polygon.

In the following example, the right box provides the double-faced polygons corresponding to the polygons in the left box.

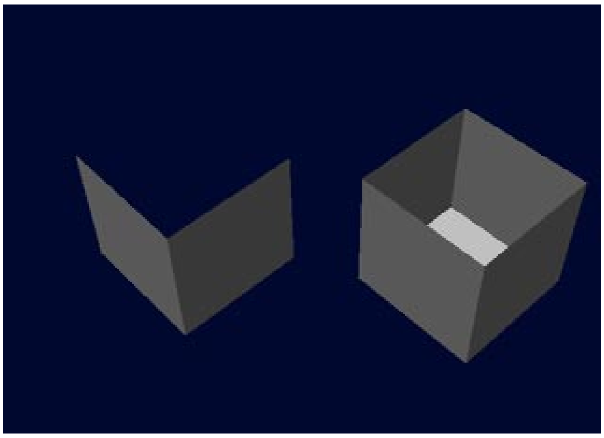


Figure 3-9-8 Double-faced Polygon

Reversing button

Reverses the selected polygon.

This function is applied to reversed model data polygons and the creation of double-faced polygons (see paragraph for the copying function).

Rendering color cataloging button

The current rendering color is cataloged as the polygon material, with polygon light source calculation turned off. For example, if a white polygon subjected to red light appears pink, the pink is cataloged as the polygon color.

This function allows the three-dimensional appearance of the model, even though it is not subjected to light source calculation on the program. The rendering color writing of a polygon through smooth shading after the appropriate setting of the light source leads to the creation of a model with very effective gradation color.

The material of a polygon where a rendering color has been written is changed as follows:

Before writing	**After writing**
Flat color	Flat color
Smooth color	Gradation color
Flat texture	Color texture
Smooth texture	Gradation texture

Rendering color clear button

Resets the material to the status before writing the rendering color.

With a rendering color written, this function cannot be used.

Polygon selection button

Clicking this button sets up the polygon selection mode.

The polygon selection mode can be left by clicking the right button.

The mouse cursor is displayed on the video monitor.

Smooth shading makes it difficult to select a polygon. In the polygon selection mode, however, all polygons are rendered by flat shading.

To set up the previous shading method, leave the polygon selection mode and click the polygon selection release button.

Mouse operation in the polygon selection mode

Move the mouse cursor inside the polygon to be selected and click the left button.

The selected polygon becomes red. Dragging the mouse enables the selection of two or more polygons. Clicking the selected polygon as pressing the Ctrl key releases the selected polygon.

Pressing the F3 key causes all polygons to be selected.

Pressing the F3 key again causes the selected polygons to be released. (This also applies to other than the polygon selection mode.)

With polygons superposed on one another, polygons below are selected in order by successively clicking the left button without moving the mouse cursor.

Mouse/key operation	Result
Clicking the left button	Polygon selection
Clicking the right button	Termination of the polygon selection mode
Ctrl + clicking the right button	Release of polygon selection
Shift + clicking the right button	Brushing
Ctrl + Shift + clicking the right	Filling button

Mouse operation in the polygon selection mode

Selected polygon release button

Releases the selected polygons.

The F3 key provides the same function.

Close button

Closes the material dialogue box.

10 Procedure for Texture Mapping

Texture can be mapped by parallel projection.

1) After having selected a polygon to be provided with texture, click the “texture” radio button in the material setting dialogue box.

2) The file dialogue box pops up. Select the TIM file for texture application.

3) The “texture location” dialogue box pops up. Enter the texture image position on the VRAM, and the pallet position.

4) The “texture mapping” dialogue box pops up. The selected texture is displayed semitransparently at the top left on the video monitor.
   Adjust the positions of the selected polygon and the texture by model movement, texture movement, or texture extension or reduction.
   The selected polygons that falls in the valid range of texture mapping are displayed in yellow. Checking the valid range as moving the texture by pixel allows the texture and the polygon to be adjusted by pixel.

Texture mapping dialogue box

The dialogue box is used for the parallel displacement, and the expansion and reduction of texture.

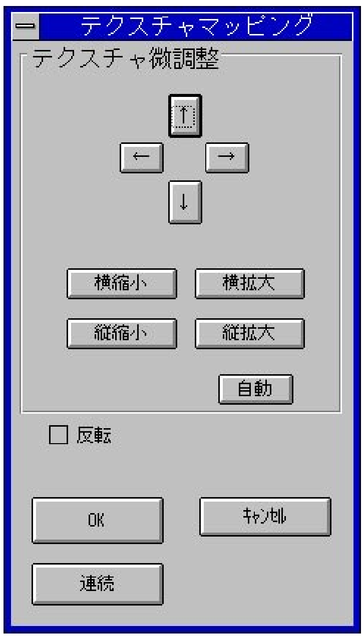


Figure 3-10-1 Dialogue Box for Texture Mapping

“→” “←” “↓” and “↑” buttons

These buttons are used for the parallel displacement oftexture by pixel. Clicking the button with the shift key pressed causes texture to be moved by ten pixels.

“Longitudinal extension” and “lateral reduction” button

The button is used for the lateral extension and reduction of texture by pixel. Clicking the button with the shift key pressed allows the extension or reduction of texture by 10 pixels.

These operations can also be carried out by the arrow keys on the keyboard.

←	Left movement	→	Right movement
↑	Upper movement	↓	Lower movement
Ctrl-←	Lateral reduction	Ctrl-→	Lateral extension
Ctrl-↑	Lateral reduction	Ctrl-↓	Longitudinal extension

Figure 3-5-2 Texture Operation by Keyboard

“Automatic” button

This button is used to automatically adjust the position and size of texture according to the selected polygon.

“Inversion” check box

The check box allows mapping with texture inverted laterally.

“Continuous” button

To apply the current texture onto another polygon, click this key instead of the OK key. The semitransparent target texture is left on the video monitor for continuous texture mapping.

“OK” button

Clicking this button leads to texture application, followed by the closing of the texture mapping dialogue box.

[Application]

Multi-pallet texture

Assigning one TIM image two or more different pallets enables the same images to be colored variously. This means allows images in TIM data to be shared. Thus, the VRAM can be used effectively. TIM can describe more than one pallet (multi-pallet). But the RSD format can index only the start address in two or more pallets. Thus, each polygon cannot be assigned a pallet.

Possible actions are to create two or more TIM files with different pallet addresses from a TIM file for a multi-pallet, and read in such TIM files through the material editor. The TIM data is stored in different files. As image addresses and data contents are the same, however, the TIM data provides the same effects as a multi-pallet on the VRAM.

The following example describes such actions.

1) Use the sprite editor to create a multi-pallet TIM file (FOO.TIM).

2) Us the TIMULTIL.EXE pallet writing function to save pallets into different TIM files.

Example)	Pallet	Pallet address	File
	0	(480, 0)	FOO0.TIM
	1	(481, 0)	FOO1.TIM
	2	(482, 0)	FOO2.TIM
	…	…	…
	N	(480+N, 0)	FOON.TIM

3) Read in FOO0.TIM.FOO1.TIM,... as texture from the material editor.

11 Light Source Menu

The light source setting dialogue box pops up.
Set a light source, an ambient color, and a background color.

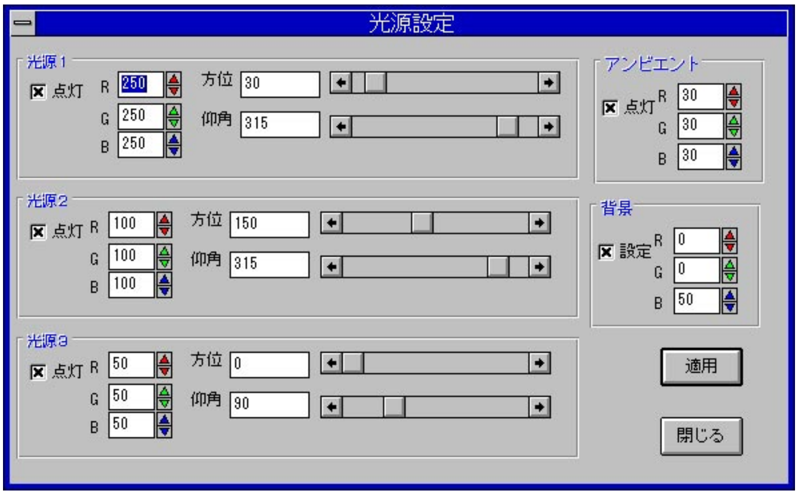


Figure 3-11-1 Light Source Setting Dialogue Box

“Lighting” check box

This check box allows the light source to be turned on and off.

RGB

Sets the color of the light source.

Bearing and elevation

Sets the direction of the light source.

Ambient

Sets an ambient color.

Background

Sets the color of the background.

12 Move Menu

The “visual point/model movement” dialogue box pops up.
Set methods of model movement, visual point movement and rendering.

Figure 3-12-1 Visual Point/Model Movement Dialogue Box

“Movement of visual point”

Clicking the button allows the parallel displacement and rotation of the visual point.

The model and the visual point can also be moved by keyboard operation.

Model rotation

A:	Left	S:	CW distortion
D:	Right	C.	CCW distortion
W:	Upper		
X.	Lower		

Parallel displacement of visual point

Shift-A:	Left	Shift-S:	Away from yourself
Shift-D:	Right	Shift-C:	Toward yourself
Shift-W:	Upper		
Shift-X:	Lower		

Model rotation (Units of 90 degrees)

Ctrl-A:	Left	Ctrl-S:	CW distortion
Ctrl-D:	Right	Ctrl-C:	CCW distortion
Ctrl-W:	Upper		
Ctrl-X:	Lower		

Note

Shortening “period until automatic repetition start” and “repetition interval” on the keyboard provides smooth model movement.

The keyboard command on the Windows control panel can be used for this setting.

“Distance between visual point and screen”

Sets the distance between the visual point and the screen. The picture angle is changed according to this value.

“Zoom”

Extends the screen display without moving the visual point.

"Rendering"

Sets a method for rendering a model. The set rendering method is not reflected in material data.

❍ Wire frame

Displayed in a wire frame.

❍ Texture mask

Displayed with no texture applied.

❍ Light source calculation not carried out

Displayed with light source calculation not carried out.

❍ Default

Displayed according to the model material.

13 Vertex Edit Menu

The "vertex edit" dialogue box pops up. The dialogue box is used to move the vertices of a model. It can also be used to specify a vertex color for gradation color polygon creation.

"Vertex ID"

The ID of the selected vertex is displayed.

"Vertex selection" button

Clicking this button sets up the "vertex edit mode", with the mouse cursor appearing on the video monitor.

Clicking the left button with the mouse cursor located near a vertex causes the vertex to be selected. The selected vertex is displayed by a blue triangle.

To leave the vertex selection mode, click the right button

Mouse operation in the vertex selection mode

Drag and move a vertex by the mouse. Moving the mouse to the right causes the vertex to be moved along the screen.

With the Ctrl key pressed, the vertex is moved only vertically.

With the Shift key pressed, the vertex is moved only horizontally.

With the Ctrl and Shift keys pressed, the vertex is moved only toward or away from yourself.

[Application]

The use of the vertex edit function along with the vertex read function enables the UV value to be adjusted finely during texture mapping. The following example maps a square texture onto a trapezoid area.

1) Use the vertex edit function to make the trapezoid area rectangular.

2) Map the texture correctly in accordance with the rectangular area.

3) The model remains rectangular. Use the vertex read function to load the PLY file for the original model to make the rectangular area trapezoid.

Now, the square texture is mapped correctly onto the trapezoid area.

Creating gradation color polygon

Setting different colors for the vertices of a polygon allows the creation of the polygon with gradation color.

1) Select a polygon to be provided with gradation color.

2) Select vertices for which colors are to be set.

3) Use the “color” radio button in the material dialogue box to select vertex colors.

14 Group Menu

The “group” management dialogue box pops up.

The dialogue box allows a polygon group to be managed.

Only group selection enables the selection of all the polygons belonging to the group at a time. Thus, materials can be edited rapidly.

“Polygon count”

Number of polygons selected

“Cataloging” button

Clicking this button causes the “group name” dialogue box to be displayed. Clicking the OK button with a group name specified in the dialogue box allows the currently selected polygon to be cataloged as a group.

“Group list”

Lists the names of cataloged groups and the number of polygons.

Clicking a group allows the selection of all the polygons belonging to the group.

“Delete” button

Deletes the selected group.

“Automatic creation” button

Automatically creates a group for each model material.

“Rename” button

Renames the selected group.

15 OT Menu

The “ordering table” dialogue box pops up.

The dialogue box is used to set the bit length of the ordering table (1 to 14 bits).

Larger bit length provides high accuracy of polygon depth judgment. But the rendering speed is lowered.

Figure 3.15-1 odtlg.pict OT Dialogue Box

16 Resolution Menu

Sets the resolution of the video monitor.

When setting screen resolution, take texture location on the VRAM fully into account. Replacing screen resolution of 320 x 240 for the current texture location by resolution of 640 x 240 may result in the destruction of part of the texture on the VRAM.

Figure 3.16.1 resoluti.pict Resolution Dialogue Box

17 Help Menu

Displays methods of binding keys, and operating the keyboard and the mouse.
Displays how to use the help menu.
Displays version information and the version of the program currently used.

CHAPTER 4

ANIMATION TOOL (ANIMATIO.EXE)

This chapter briefly sets out the files handled by the animation tool (Ver.1.X), and explains methods of operating dialogue boxes.

1 Outline

1.1 Functions

This software is an authoring tool dedicated to motion design. It does not support modeling and rendering functions. Checking an animation provided with textures requires hardware DTL-H2000, authoring tool TOD View, or a programmer tool.

The animation tool supports the following functions.

- Layout of an object on a three-dimensional space
- Definition of the hierarchical structure of an object
- Creation of animation data
- Creation of information for use in data identification in the PlayStation unit
- Import of files
  Startup of the command for converting a DXF file into the RSD file
- Export of files
  Output of animation data
  Startup of the command for converting an RSD file into TMD data.
  Startup of the output command of hierarchical data

1.2 File Configuration

This subsection outlines files handled by the tool.

Internal files

This file is created and edited by the tool.

- **PRJ file**

Extension: .PRJ
File contents:
OBJECT.PRJ
Information used to identify an object in the PlayStation unit
MODEL.PRJ
Information used to identify TMD data in the PlayStation unit

- **HRC file**

Extension: .HRC
File contents:
Include hierarchical data for a PC, and information for defining relationships between objects laid out. This file stores a list of sequence names cataloged by HRC.

- **ACT file**

Extension: .ACT

File contents:

Include sequence data for a PC (set of arranged key frame data), and information for use in defining the position and direction of a created object.

- **RSD file**

Extension: .RSD

File contents:

Management of files storing information for defining model shapes and colors for a PC

◇ External files

The external file storing model data to be converted into internal files and animation data to be passed to the PlayStation unit cannot be handled directly by the tool.

- **DXF file**

Extension: .DXF

File contents:

Model data created by modelers of other firms. This data is converted by the Import command into the corresponding RSD file to be used.

- **TMD file**

Extension: .TMD

File contents:

Model data for the PlayStation unit

The Export command is used for conversion from an RSD file.

- **TOD file**

Extension: .TOD

File contents:

Include hierarchical data or sequence data for the PlayStation unit converted by the Export command from an HRX file and from sequence data on the memory.

1.3 Project

This tool first creates a project to uniquely manage an object. For example, a game tile is allocated as a project. This prevents the duplicate occupation of the PlayStation memory by the same object in the game.

A project refers to a total item composed of the following directories and files, and does not refer to a certain file,

A project is prohibited from looking into and include part of another project.

```
(project directory)
  |
  |--- model.prj file
  |--- object.prj file
  |
  |--- act directory
  |       └── act files
  |--- cam directory
  |--- hrc directory
  |       └── hrc files
  |--- lht directory
  |--- org directory
  |--- rsd directory
  |       └── act files
  |--- tim directory
  |--- tmd directory
  |       └── tmd files
  |--- tmp directory
  └─── tod directory
          └── tod files
```

1.4 Required Environment

Operation and actual running require the environment which satisfies certain specifications.

The empty capacity of the needed RAM and HD depend significantly on the other applications running on Windows, the network driver, and set CONFIG.SYS, AUTOEXEC.BAT, and WIN.INI, as well as on the data used, however. This, the tool may not be operated even though the following values are met.

DOS/V:	COP Intel486DX2 66 MHz
RAM:	16 MB
HD:	120 MB
	Inner virtual memory: 35 MB
Video:	17 inches
	256 colors available concurrently
	Resolution: 1024 x 768 dots
OS:	DOS version 5.0 or later versions
	Microsoft Windows 3.1
Others:	◇Mouse
	◇Keyboard
	◇Visual Basic Dynamic Link Library (DLL)

2 Menus/Commands

2.1 Project Menu

New Project - Creates a new project.

Operation

① Select New Project.
The "New Project" dialogue box is displayed.

② Specify a drive and directory for project creation.

③ Enter the project name and click the "Create" button.

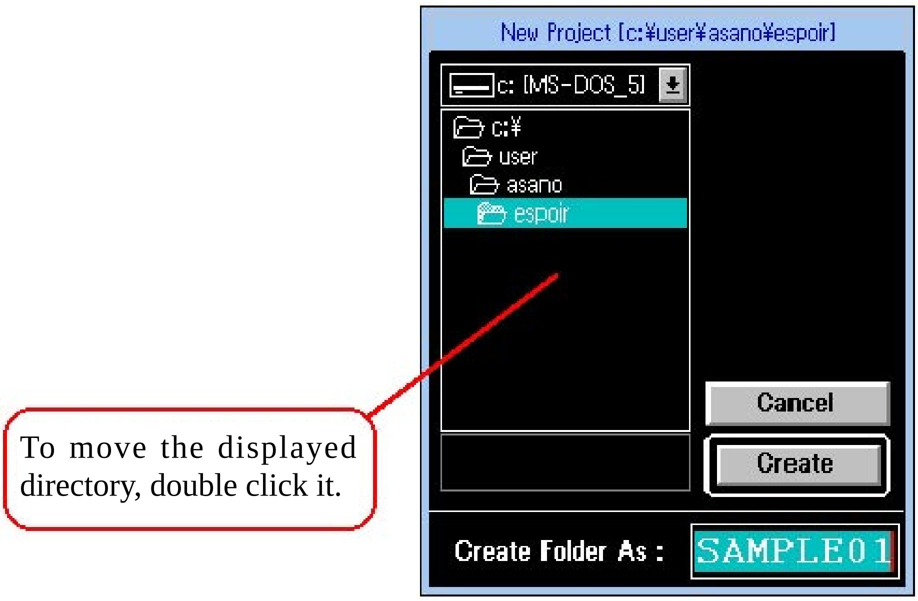


Figure 4-2-1 New Project Dialogue Box

Open Project - Opens a project.

Cataloging a new file into or editing a cataloged file in the desired project requires that the project be opened.

Operation

① Select Open Project.

② Moves the open project to the project directory.

③ Select the project and click the “Open” button.

Note

In the following Figure, sample() is the project directory. Such project files as model.prj and object.prj need not be selected.

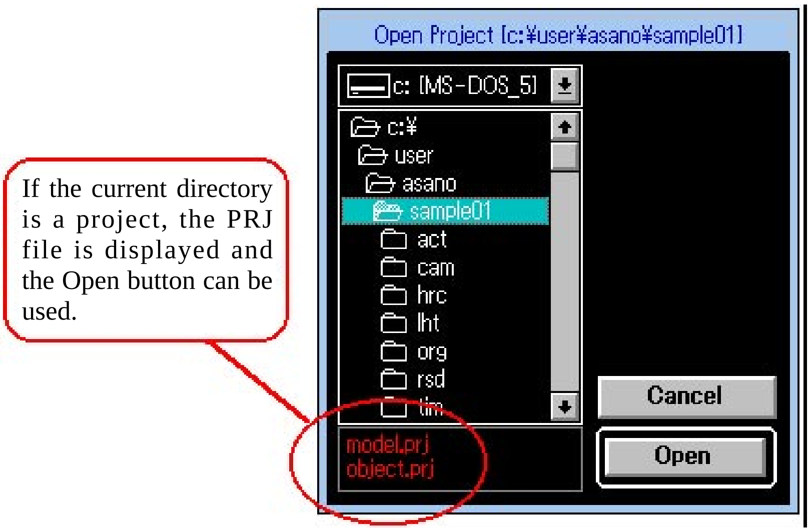


Figure 4-2-2 Open Project Dialogue Box

Close Project - Closes a project.

Operation

Select Close Project.

Note

The dialogue box for checking that the HRC file is saved is not displayed.

If this command has been issued with the new sequence opened, the dialogue box for asking whether to save the sequence is displayed. In that case, the HRC file as well as the ACT file is saved. This is because the sequence name is cataloged into the HRC file and because the HRC and ACT files are adjusted for the added object and the new hierarchical structure from the opening of the HRC file until the storage of the sequence. Note that the position and attitude of an object written in the HRC file are also updated.

If the tool has detected the absence of the key frame during ACT file storage, the dialogue box for requiring the deletion of the ACT file is displayed.

New Sequence - Creates a new sequence.

The New Sequence dialogue box is used to set a sequence name. Saving the HRC file leads to the cataloging of its name. The name plus extension ".ACT" is the name of the ACT file.

Operation

① Select New Sequence.
The "New Sequence" dialogue box is displayed.

② Enter a sequence name and click the "Create" button.
"Keyframe Recorder" is displayed.

Note

The sequence names cataloged in the currently opened HRC file are displayed in gray characters in the "New Sequence" dialogue box. The name of the new sequence to be created must not be the same as any of those names.

This command requires that a sequence name be cataloged into the HRC file. Thus, it cannot be executed if the HRC file has not been saved after the start of a job by the New command in the File menu.

Remarks

The entered sequence name is displayed on the "Keyframe Recorder" title bar.

Open Sequence - Opens the ACT file.

Operation

① Select Open Sequence.
The “Open Sequence” dialogue box is displayed.

② Select a sequence and click the “Open” button.
“Keyframe Recorder” is displayed.

Remarks

The selected sequence name is displayed on the “Keyframe Recorder” title bar.

Close Sequence - Closes a sequence.

Operation

Select Close Sequence.

Note

The dialogue box for asking whether to save an updated sequence is displayed. In that case, the HRC file as well as the ACT file is saved. This is because the sequence name is cataloged into the HRC file and because the HRC and ACT files are adjusted for the added object and the new hierarchical structure from the opening of the HRC file until the storage of the sequence. Note that the position and attitude of an object written in the HRC file are also updated.

If the tool has detected the absence of the key frame during ACT file storage, the dialogue box for requiring the deletion of the ACT file is displayed.

Save Sequence - Saves a sequence into the ACT file.

Operation

Select Save Sequence.

Note

The HRC file as well as the ACT file is saved. This is because the sequence name is cataloged into the HRC file and because the HRC and ACT files are adjusted for the added object and the new hierarchical structure from the opening of the HRC file until the storage of the sequence. Note that the position and attitude of an object written in the HRC file are also updated.

If the tool has detected the absence of the key frame during ACT file storage, the dialogue box for requiring the deletion of the ACT file is displayed.

2.2 File Menu

New - Create a new HRC file.

Operation

Select New

The General window is displayed. Normally, the window covers coordinate axes of X (red), Y (green) and Z (blue) for an applicable three-dimensional space (word coordinate system).

Note

Strictly speaking, no HRC file is created if the following commands are not executed.

Save command
Save As command
Save Sequence command (including an answer of saving in response to the dialogue box for saving the new sequence)

Open - Opens the HRC file.

Used to invoke and modify the saved HRC file.

Operation

① Select Open. The “Open” dialogue box is displayed.

② Select the file and click the “Open” button.

Add Object - Adds a model to the applicable three-dimensional space.

When including a model into the applicable three-dimensional space, first select the RSD file of the model to be added. Then, name the model. (This tool identifies an object by the name. The set name allows object selection.)

The loaded model is a member of the applicable three-dimensional space (world coordinate system). (For the hierarchical structure, see “Link tool” in the Tool menu.)

Operation

① Select Add Object.
The “Add Object” dialogue box is displayed. (Left part of the Figure below)

② Select the RSD file (model) to be loaded in “Source List Box”.
Now, the “Add” button can be used.

③ Click the “Add” button.
The selected RSD file is displayed in “Add List Box”.

④ Click the “Done” button to determine the file to be loaded.
The “Name Table” dialogue box is displayed. (Right part of the Figure below)

⑤ Enter an object name into the Object Name column.

⑥ Click the “Set” button to load the file.
The added model is displayed in the wire frame.

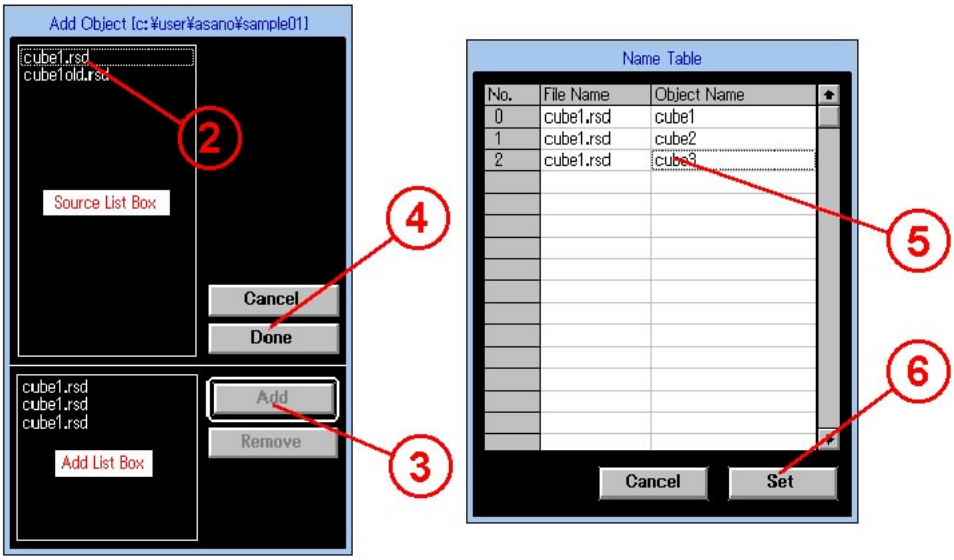


Figure 4-2-3 Add Object and Name Table Dialogue Boxes

Add Object dialogue box

- **Add button**

Displays a file name selected from “Source List Box” in Add List Box under the dialogue box.

Remarks

Double clicking the file to be selected in “Source List Box” makes it unnecessary to operate the “Add” button.

- **Remove button**

Cancels the file selected in Add List Box. Select the file to be canceled and click the “Remove” button.

- **Done button**

Fixes the selected file. Clicking this button causes the “Name Table” dialogue box to be displayed.

Name Table dialogue box

- **File Name box**

The selected file name is displayed.

- **Object Name box**

The object name of the selected file is typed. The selected files must be assigned different object names.

Two or more files cannot be assigned the same object name.

- **Set button**

Catalogs the RSD file name into the Model.PTJ file and the object name into the OBJECT.PRJ file.

Important!

Each object name is unique. Even though the cataloged object has been deleted from the HRC file, its name is saved permanently.

The object name can be entered in either uppercase or lowercase characters. Box, BOX and box are considered to be the same object name.

The reserved words include the object names of default cameras.

IsometricCamera
TopCamera
FrontCamera
LeftSideCamera

Close - Closes the HRC file.

Operation

Select Close.

Note

The dialogue box for checking that the HRC file is saved is not displayed.

If this command has been issued with the new sequence opened, the dialogue box for asking whether to save the sequence in the ACT file is displayed. The HRC file as well as the ACT file is saved. This is because the sequence name is cataloged into the HRC file and because the HRC and ACT files are adjusted for the added object and the new hierarchical structure from the opening of the HRC file until the storage of the sequence. Note that the position and attitude of an object written in the HRC file are also updated.

If the tool has detected the absence of the key frame during ACT file storage, the dialogue box for requiring the deletion of the ACT file is displayed.

Save - Saves the HRC file.

Operation

Select Save.

Save As - Saves the HRC file under an alias.

The new HRC file created by New or the currently opened HRC file is saved under an alias.

Operation

① Select Save As.
The “Save As” dialogue box is displayed.

② Enter a file name and click the “Save” button.

Get Info - Displays details of the selected object.

Operation

Select an object and, then, Get Info. The “Get Info” dialogue box is displayed.

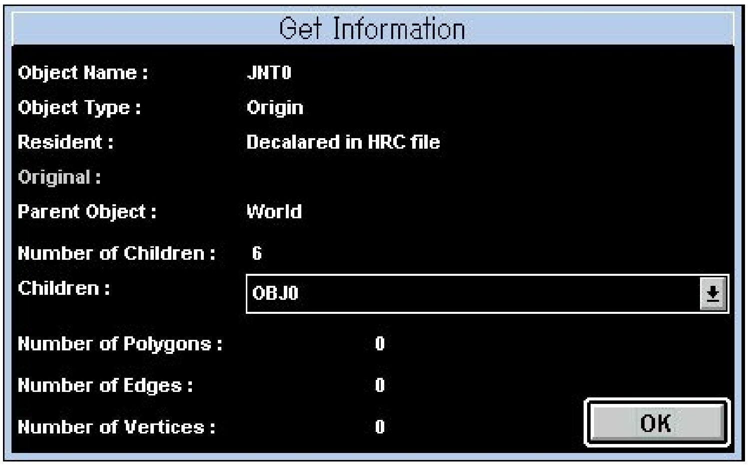


Figure 4-2-4 Get Information Dialogue Box

- **Object Name**

  Displays the object name specified by the following commands.

  Model: Add Object command
  Camera: Create Camera command
  Origin: Create Origin command

- **Object Type**

  Displays the type of an object (model/camera/origin).

- **Resident**

  Displays the RSD file path for a selected object model. For a camera/origin, “Declared in HRC file” is displayed as shown above.

- **Original**

  This flag is not used currently.

- **Parent Object**

  Displays a parent object name. (For parenthood, see “Link Tool” in subsection 2.2.8, Tool Pallet.)

- **Number of Children**

  Displays the number of set member objects.

- **Children list box**

  Displays a member object name.

- **Number of Polygons**

  Displays the number of polygons constituting a selected object model.

- **Number of Edges**

  Displays the number of edges constituting a selected object mode. This value is an estimated value.

- **Number of Vertices**

  Displays the number of vertices constituting a selected object model.

Import - Converts a DXF file into the corresponding RSD file.

This tool can convert the model developed by commercially available CG software into the RSD file by starting up DXF2RSD.EXE. However, this requires that the model is saved by the software as a DXF file that is supported by DXF2RSD.EXE supplied by our firm.

The RSD file created through the conversion is stored into the RSD directory of the object opened during the startup of the conversion.

Important!

The use of this command involves the installation of the DXF2RSD.EXE file in directory $PSXGRAPH$BIN for drive C:.
This command is not operated if the file is not installed in a different directory.

Note

This command can be used after the directory of an opened project for storing the results of conversion has been fixed. Therefore, it can be used even with the HRD file opened. But the created RSD file is not included in the HRC file. Including the created RSD file in a certain HRC file calls for using the Add Object command.

Remarks

The use of DXF2RSD.EXE on the DOS or of its Windows version, DXF2RSDW.EXE, allows finer setting for conversion.

Operation

① Select Import. The “Import” dialogue box is displayed.

② Target the drive and directory storing the DXF, and select a file to be converted.

③ Enter a file name to be used after conversion.

④ Set parameters.

⑤ Click the “Import” button.

Note

Upon the start of the processing, the black DOS screen is set up. Following the DOS screen, however, the Windows screen pops up.

The RSD file is not converted. The existing RSD file to be used is to be copied into the RSD directory using the file manager.

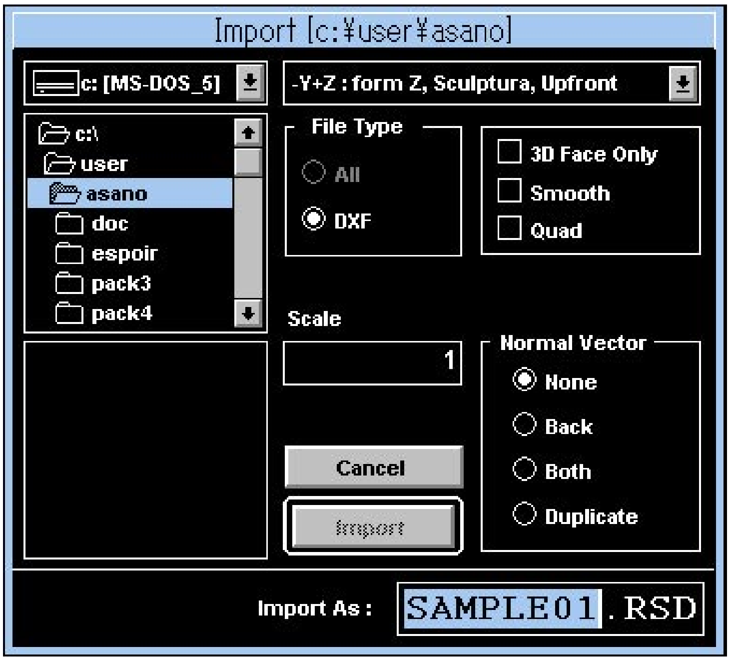


Figure 4-2-5 Import Dialogue Box

- **Coordinate System list**

Specifies a method for the transformation of the DXF coordinate system. Specifies the upper coordinate axis and its direction, as viewed from the front. For example, "-Y+Z" indicates that the Y axis is negative toward yourself and the Z axis is positive toward the top. The coordinate system for DXF does not always coincide with the modeler screen. A test should be carried out if the appropriate software name is displayed.

- **File Type**

Specifies a file type to be displayed in a file list.

All: Displays all files on the current directory.
DXF: Displays only the files having an extension of ".DXF".

- **3D Face Only check box**

Ignores POLYLINE and converts only 3DFACE.

- **Smooth check box**

Carries out smooth shading.

- **Quad check box**

Does not divide quadrangular 3DFACE into triangles. This allows reduction in the number of polygons for a whole model.

- **Scale box**

Used to extend a model to be converted.

- **Normal Vector radio button group**

None: Does not carry out the following processing.
Back: Reverses the directions of the normals for all polygons.
Both: Creates all polygons as double-faced polygons.
Duplicate: Creates the obverses and reverses of all polygons independent of one another.

- **Import button**

Uses the set parameter to start up DXF2RSD.EXE.

- **Import As text box**

Enters the name of a file created by conversion. The new RSD file is saved into the RSD directory of the currently opened project.

Supplementary information: Model data conversion

The following example shows the process of converting a DXF file, displaying the model and saving the HRC file.

① Start up this software.

② Select the New Project command in the Project menu to create a project.

③ Select the Import command in the File menu for conversion.

④ Select the New command in the File menu to create a space for application.

⑤ Select the Add Object command in the File menu to add the created RSD file to the application space.

⑥ Select the Save As command in the File menu to save the HRC file.

The following gives actions to be taken when no model is displayed.

First, repeatedly press the space key to near the origin of the application space. If a black point appears as nearing the origin, this indicates that the model is too small for the PlayStation unit. Carry out conversion again. Be sure to specify a larger value in the Scale column in the dialogue box.

If none appear although the origin is neared, refer to the manual for DXF2RSD .EXE to check the CG software. Even though the software is applicable, the model may not be displayed because of:

- ❍ parameters for creating the DXF file;
- ❍ a DXF release version; or
- ❍ parameters for conversion.

Export TOD - Creates a TOD file.

A TOD file can be created by converting only hierarchical structure or both hierarchical structure and sequence. If the moving pattern is strong as in RCUBE in a demo for the programmer or if such items as BALLS having no pattern are required, only hierarchical structure is converted as the object must be moved by a program. Using the sequence created by this tool to generate an animation involves the conversion of both hierarchical structure and sequence.

Important!

Using this command to convert only hierarchical structure requires that the MKTOD.EXE file be installed in directory ¥PSXGRAPH¥BIN for drive C:. If the file is installed in a different directory, this command is not executed.

Operation

◇ Conversion of only hierarchical structure

① Select an object to be converted. This selection is not required when all objects are converted.

② Select Export, and "TOD" in the cascade menu. As shown below, the "Export TOD" dialogue box is displayed.

③ If operation in step ① has been carried out in the Export Object column, Selected can be selected. If not, or when converting all objects, select All.

④ In the File Type column, select Structure.

⑤ Enter a file name to be used after the conversion, and click the "Export" button.

◇ Conversion of both hierarchical structure and sequence

① Select an object to be converted. This selection is not required when all objects are converted.

② Select Export, and "TOD" in the cascade menu. As shown below, the "Export TOD" dialogue is displayed.

③ If operation in step ① has been carried out in the Export Object column, Selected can be selected. If not or when all objects are converted, select ALL.

④ In the File Type column, select Animation. The "Export TOD" dialogue is extended and displayed as shown below.

⑤ Set parameters for an animation.

⑥ Enter a file name to be used after the conversion, and click the "Export" button.

Figure 4-2-6 Export TOD Dialogue Box

- **Export Object**

Specifies an object to be converted.

All:
Converts all objects depending on the file type.

Selected:
Converts only the selected objects. With no object selected, this button cannot be selected.

- **File Type**

Specifies the type of a file to be converted.

Structure: Converts only hierarchical structure into the TOD file.

The currently opened HRC saved last is converted into the TOD file. Information not saved is not converted. Clicking this button inverts the status.

Animation: Converts both hierarchical structure and sequence into the TOD file.

Data on the memory is converted into the TOD file. The data includes the currently opened HRC file, the ACT file, and operation results not saved. However, the objects for which no significant key frame is created, and the default camera are not converted. Clicking this button inverts the status.

- **Export button**

Starts conversion.

The following explanation is applicable only with Animation specified for File Type.

- **Start[Frame]**

Specifies the start frame of the created sequence to be saved. The PlayStation unit fetches objects from the CD-ROM into the memory to reproduce animations. The objects which have been fetched into the memory need not be fetched again into the memory. Sequence reproduction calls for specifying whether the object to be converted have been fetched into the memory.

Create: Converts the object to be fetched first into the memory.

Remarks

Create object control packets are developed for all the frames specified by Start. Given below is a description of the color of the key frame marker with the key frame set in the frame.

Green: Create object control packet
Red: Create object control packet
Yellow: Create object control packet
Not set: Create object control packet

Exists: The object already fetched into the memory is converted.

Remarks

Given below is a description of the color of the key frame marker with the key frame set in the frame specified by Start.

Green: Create object control packet
Red: Kill object control packet
Yellow: No packet
Not set: No packet

- **End[Frame]**

Specifies the end frame of the created sequence to be saved.

- **Offset[Frame]**

Specifies the frame at which operation is to be started after the sequence has been invoked.

- **Frame Rate[Sec/Frame]**

Specifies the period for the reproduction of one frame.

- **Export As text box**

Specifies a file name to be used after the conversion.

Supplementary information:
Conversion of hierarchical structure (File Type = Structure)

The HRC file on the hard disk is processed by this command.

With the New command in the File menu used to create a new HRC file, no HRC file is created in the hard disk. The file must be saved immediately before this command is executed.

Further, if the “Open” command in the File menu has been used to open the HRC file to be edited, the file must be saved immediately before this command is executed. If the file has not been saved, the results of editing are not reflected in the HRC file.

Upon the start of the processing, the black MS-DOS screen set up. Upon the end of the processing, however, the Windows screen is set up.

Supplementary information: Conversion of hierarchical structure and sequence (File Type = Animation)

The sequence can be converted after the New Sequence or Open Sequence command in the Project menu has been executed until the execution of Close Sequence.

Export TMD - Creates a TMD file.

Important!

The use of this command involves the installation of the RSDLINK.EXE file in directory ¥PSXGRAPH¥BIN for drive C:.

If the file is installed in a different directory, this command cannot be executed.

Operation

① Select a model to be converted. This selection is not required when all the models contained in the HRC file are converted.

② Select Export, and “TMD” in the cascade menu. The “Export TMD” dialogue box is displayed.

③ If operation in step (1) has been carried out in the Export Model column, Selected can be selected. If not or when all objects are converted, select All.

④ Enter a file name to be used after the conversion, click the “Export” button.

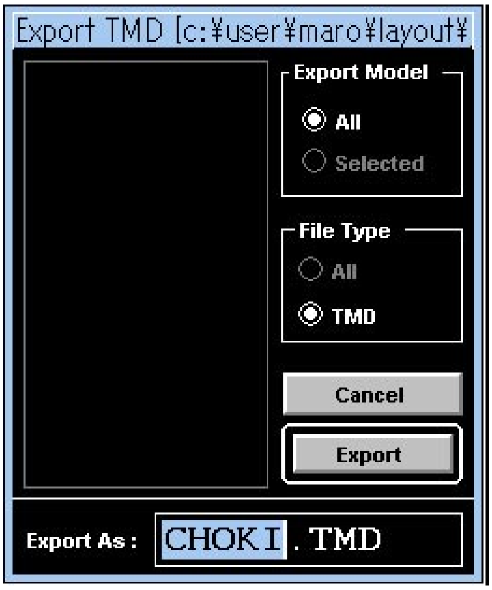


Figure 4-2-7 Export TMD Dialogue Box

- **Export Model**

  Specifies a model to be converted.

  All: Converts all models.
  Selected: Converts only the selected model. If no model has been selected, this button cannot be selected.

- **File Type**

  Specifies the type of a file to be converted.

  All: Not used
  TMD: Converts a file into the TMD format. Information on the color and texture of model data is also converted.

- **Export button**

  Converts the selected model into the TMD format.

- **Export As text box**

  Specifies a file name to be used after the conversion.

Supplementary information: Model data conversion

Model data can be converted when the model is on the memory, i.e. after the New or Add Object command in the File menu has been executed or when the HRC file has already been opened by the Open command in the File menu.

Upon the start of the processing, the black MS-DOS screen is set up. Upon the end of the processing, however, the Windows screen is set up.

After execution, a header file (with extension of .h) containing a list of RSD ID numbers is created in the Project directory. The header file is needed for the programmer.

Quit - Terminates the tool.

Operation

Select Quit. The tool is terminated.

Note

If this command has been issued with the updated sequence opened, the dialogue box for asking whether to save the ACT file is displayed. The HRC file as well as the ACT file is saved. This is because the sequence name is cataloged into the HRC file and because the HRC and ACT files are adjusted for the objects added after the HRC file has been opened until the sequence is saved and for the updated hierarchical structure. Note that the position and attitude of the object written in the HRC file are also updated.

If this tool has detected the absence of the key frame during the saving of the ACT file, the dialogue box for requiring the ACT file to be deleted is displayed.

2.3 Edit Menu

Undo/Redo - Undoes the last executed operation.

Operation

Immediately after the menu or the tool has been used, select Undo. The operation is canceled, with the menu item changed to Redo. Selecting Redo leads to the setup of the previous status.

Supplementary information: What can be done by Undo

The following gives what can be done by Undo.

◇ Parallel displacement, rotation, extension and reduction of an object

◇ Hierarchical structure: Link and Unlink commands

◇ Lock and Unlock commands

◇ Cut, Copy and Paste command

Note

Undo by Ctrl-Z is valid, only when the window has a focus. Clicking the tool packet moves the focus to the tool palette. Thus, Undo by Ctrl-Z is unavailable. Unto under this status involves the selection of Undo in the pull-down menu.

Note that focus movement by clocking a window may cause unexpected results (except for selection by a title bar).

Operation in the key frame recorder is not covered by Undo. For example, with an object rotated to catalog the key frame, the object is reset by Undo, but the key frame is not deleted.

Cut - Erases the selected object for inclusion in the clip board.

The selected object disappears from the screen.

Operation

Select the object to be included into the clip board and, then, Cut.

Copy - Includes a copy of the selected object into the clip board.

The selected object is left on the screen.

Operation

Select the object to be included into the clip board and, then, Copy.

Paste - Copies the object included in the clip board.

Copies the object included by Cut or Copy into the clip board.

Operation

Select Paste. The “Name Table” dialogue box is displayed.
Enter an object name into the New Object Name box and click the “Set” button.

Remarks: “Name Table” dialogue box

The name of the last pasted object is displayed in the Old Object Name column in the dialogue box.

Supplementary information: Cut, Copy, Paste

Objects and their member sub-objects are covered by Cut, Copy and Paste. Copying the selected shoulder object leads to the duplication of the shoulder and the arm. Before a hierarchy is created or by temporary Unlink, only the shoulder object can be copied.

Cut and Copy duplicates parameters for the rotation, extension and parallel displacement of the original object. The Paste command is used to duplicate the object in the same attitude, size and position.

For hierarchical structure, however, the parent information of the original object (selected for Cut or Copy) is not duplicated. The pasted object belongs to the world coordinate system of the uppermost layer. Naturally, the hierarchical structure below the pasted object is the same as of the original object.

The clip board is located on the memory. Cutting or copying data over 30,000 polygons consumes a large amount of memory and makes the operation unstable depending on the swap setting. In such a case, save the needed file and terminate the program. The memory benefits from loads reduced by the clip board for the subsequent startup.

The three-dimensional clip board is particular to this tool. Data cannot be exchanged with other software via the board.

Clear - Erases the selected object.

Operation

Select the object to be erased and, then, Clear.

Preferences - Initialization (Updates plane size and position.)

The position and roughness of a plane for use in movement evaluation can be set. The parameters set below are default values.

Operation

Plane Min: Specifies the plane start position.

Plane Max: Specifies the plane end position.

Grid Pitch: Specifies plane roughness.

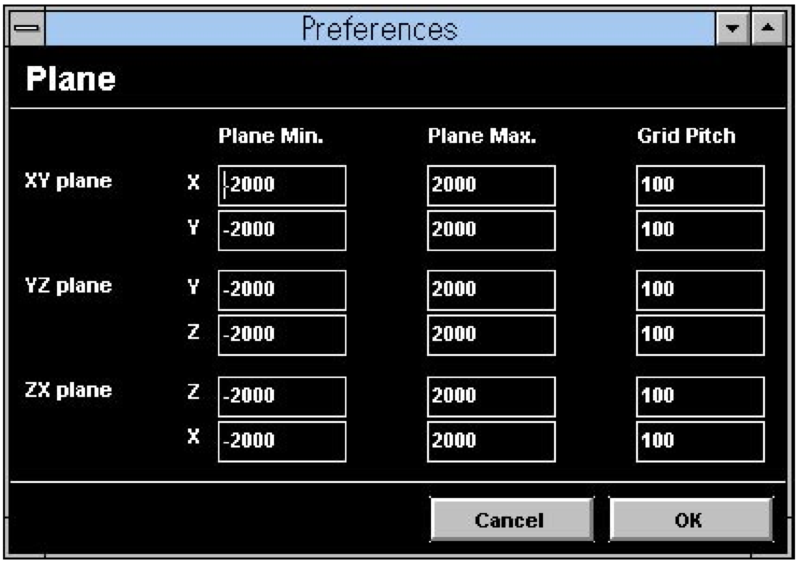


Figure 4-2-8 Preferences dialogue box

2.4 View Menu

Magnify - Magnifies and displays window contents.

Short Cut: “Magnifier tool” in the tool palette

Operation

Magnifies by a factor of two and displays window contents around the clicked position.

Minify - Minifies and displays window contents.

Short Cut: Shift key plus “Magnifier tool” in the tool palette

Operation

Minifies by a factor of two and displays window contents around the clicked position.

Supplementary information:
Magnify/Minify command and Scroll/Magnifier tool

These functions are not related to the camera. The result of operation never affects the position, direction and focus of the camera. The functions are equivalent to dark room work for printing dislocation and size changes. Therefore, the range beyond printing paper is not displayed. The range of printing paper is the square, prepared on the basis of the longer side of the full-scale window, multiplied by display power.

Show Origin/Hide Origin - Displays/hides the origin.

Operation

Select Origin. The origin is displayed, with the menu item replaced by Hide Origin.

Selecting Hide Origin hides the currently displayed origin.

Important: The origin not put in the field of view is not displayed.

The red, green and blue of the origin corresponds to the X, Y and Z axes, respectively.

Supplementary information: Origin

The origin is used to update the center of model rotation for developing hierarchical structure and connect the extended and reduced models so as to prevent its members from being affected.

The following gives differences between the origin and the model.

1. Coordinate system and origin
   The coordinate system as a mathematical concept is not the origin. Thus, the local coordinate systems for a model and a camera are not displayed.

2. The origin is not displayed under the typical status. Displaying the origin requires the selection of Show Origin in the View menu.

   Note

   The origin can be displayed by the Create Origin or Move Origin Only command as well.

3. The origin must not be extended nor reduced.
   The origin cannot be extended nor reduced directly by this tool to prevent effects of the past hierarchical structure.

   Note

   The implemented hierarchical structure of the PlayStation unit is such that extension and reduction is transmitted to descendants and cannot be canceled. Thus, during rotation, the object may frequently incur shear deformation. The tutorial provides an example of using the origin to bridge over these difficulties.

4. There is no file. There is no file equivalent to the RSD file for the model.

Show Camera/Hide Camera - Displays/hides the camera.

Operation

Select Show Camera. The camera is displayed, with the menu item replaced by Hide Camera.
Selecting Hide Camera hides the currently displayed camera.

Important!

The camera not put in the field of view is not displayed.

Supplementary information: Camera

This tool supports four default cameras used to display the General[Isometric], Top, Front and Left Side windows. But the cameras are not used for serving the PlayStation unit. Only the position and attitude information of the developed camera object are passed to the PlayStation unit.

Note

Among the default cameras, the Top, Front and Left Side cameras cannot be selected.

An attempt to select any of those cameras results in the generation of an alarm.

The following gives differences between the model and the camera.

1. The camera is not displayed under the typical status. Displaying the camera involves the selection of Show Camera in the View menu.

   Remarks: The camera can be displayed by the Create Camera or Move Camera Only command as well.

2. The camera must not be extended nor reduced.

   The object must not reflect effects of the extension and reduction of the past hierarchical structure.

3. Children of hierarchical structure must not be created.

4. More operations means are supported.

◇ **Keyboard**

With a window selected, pressing the following keys changes the display.

Left Arrow:	Facing left:	10 degrees
Right Arrow:	Facing right:	10 degrees
Left Arrow + Shift:	Tilting left:	10 degrees
Right Arrow + Shift:	Tilting right:	10 degrees
Space:	Advance:	1000
Space + Shift:	Retreat:	1000

Left Arrow + Ctrl:	Facing left:	One degree
Right Arrow + Ctrl:	Facing right:	One degree
Left Arrow + Shift + Ctrl:	Tilting left:	One degree
Right Arrow + Shift + Ctrl:	Tilting right:	One degree
Space + Ctrl:	Advance:	100
Space + Shift + Ctrl:	Retreat:	100

Note

With the camera retreating in the Back Face Cull mode, the object put in the field of view is chipped. This is because judgment on the attitude is omitted during camera movement by the keyboard for more rapid re-rendering.

◇ **See the Set View command.**

5. There is no file equivalent to the model RSD file. No focus information is saved.

Show Backface/Hide Backface - Displays/hides the reverse of an object.

Operation

Select Show Backface. The reverse of an object is displayed, with the menu item replaced by Hide Backface.
Selecting Hide Backface hides the reverse of the object. This mode is referred to as the Back Face Cull mode.

Show Plane - Displays a plane on three-dimensional space.

Related command: “Preferences” in the Edit menu

Operation

Select Show Plane and, then, any of the XY, YZ and ZX planes in the cascade menu. The selected plane is displayed.
Hiding the currently displayed plane involves the selection of Show Plane in the cascade menu.

Set View - Sets the camera to be allocated to the General window.

Used to operate or select the camera allocated to the General window.

Operation

① Select Set View.

② First, select a camera in the Camera list.

③ When changing the purse, move the Focus scroll bus to the desired position.

④ To check the view, press the Preview button. The view from the camera selected on the screen is displayed.

⑤ Click the “Set” button to select a camera.

Note

With no camera created, only IsometricCamera is displayed.

The Top, Front and LeftSide cameras are not covered.

If the camera allocated to the General window cannot be identified, check the title bar for the General window.
With no camera created, no camera is selected.

Remarks

If it is troublesome to press the Preview button, press the Space key. Double clicking in the Camera list enables both selection and inspection.

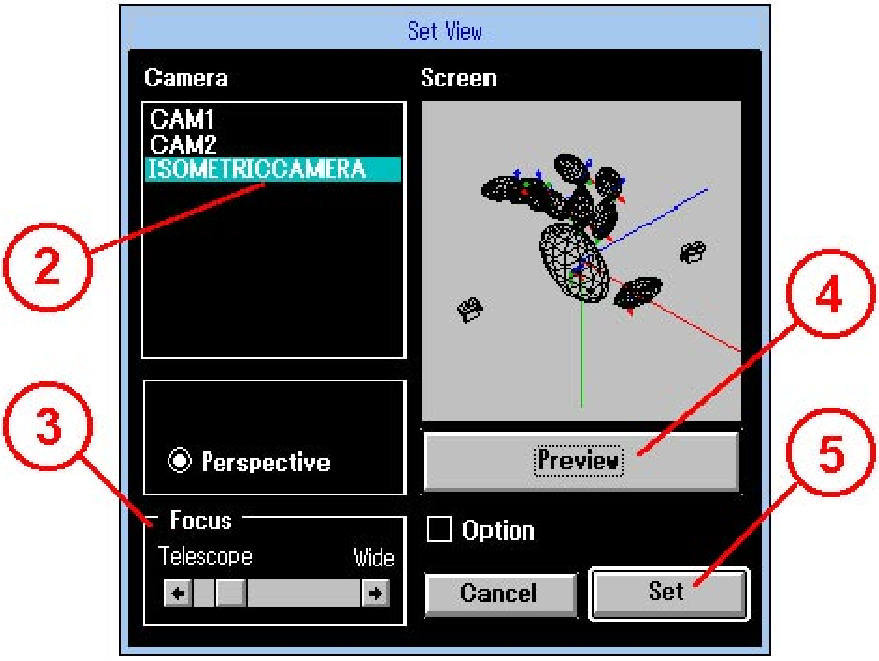


Figure 4-2-9 Set View Dialogue Box

- **Camera list**

  A list of camera names (default IsometricCamera, and any camera names assigned to the cameras created by Create Camera in the Tool menu) is displayed. The list is used to select camera to be allocated to the General window.

- **Focus**

  Modifies the focus of the selected camera.

Wide:	Wide angle
Telescope:	Telescope

- **Screen**

  Clicking the “Preview” button displays the three-dimensional space as viewed from the selected camera.

- **Preview button**

  Used to check the view from the selected camera.

- **Set button**

Fixes allocation. The view from the selected camera is displayed in the General window.

- **Option button**

Used to specify the view point, view target and twist angle (screen gradient) of the selected camera. The “set View” dialogue box as shown below is used to specify the values of parameters. In the View Point and View Target columns, red indicates the X axis, green the Y axis, and blue the Z axis.

The bottom check box is used to advance and retreat the camera by the distance specified in the right text box.

Important!

The specified value is based on the coordinate system defined by the parent object of the selected camera. The position is an absolute value, while the angle and distance are relative values.

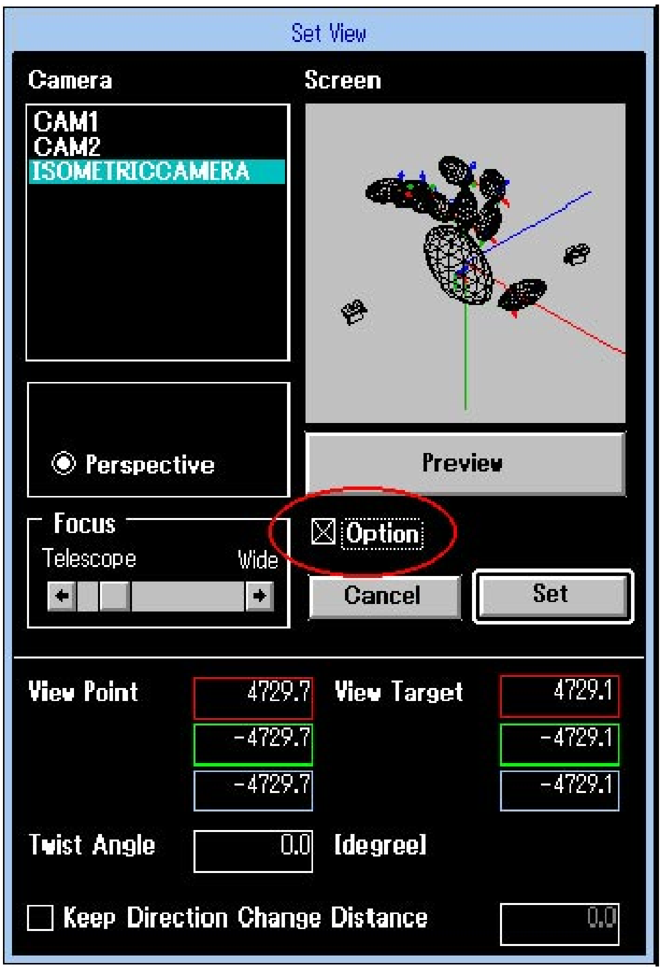


Figure 4-2-10 Set View Dialogue Box (With Option button pressed)

2.5 Tool Menu

Tool Palette - Displays/hides the tool palette.

Operation

◇ Display

Select the Tool Palette command in the menu. The left side on the command is checked.

◇ **Hide**

Re-select the Tool Palette command. The check mark disappears.
See subsection 2.2.8, Tool Palette.

Keyframe Recorder - Hides/displays the key frame recorder.

Operation

◇ **Hide**

Select the Keyframe Recorder command in the menu. The left side on the command is checked.

◇ **Display**

Re-select the Keyframe Recorder command. The check mark disappears.

Note

This command is valid with the sequence opened, i.e. after the execution of the New Sequence or Open Sequence command until the execution of the Close Sequence command. The key frame recorder is displayed first by the New Sequence or Open Sequence command.

See section 2.9, Key Frame Recorder.

Object Browser - Uses a name to select the object.

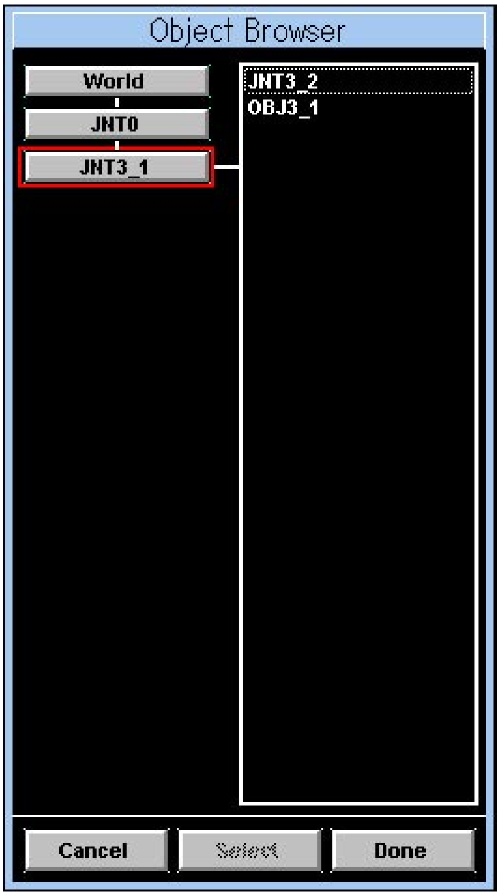


Figure 4-2-11 Object Browser

Operation

① Select the Object Browser command in the Tool menu. “Object Browser” is displayed. The buttons to the left of the browser display the selected object (indicated in the red frame. Each window displays the bounding box.), and its ancestors. The right list displays children of the selected object.

② To select an object, select an object name from the right list, and press the Select button. (Or, double click an object name in the right list.) To select an ancestor, click thee left button. To release the selection, click the World button.

③ To fix the selection, click the Done button and close the browser.

Create Origin - Creates a new origin.

Operation

① Select CreateOrigin. The “Set Origin Name” dialogue box is displayed.

② Enter the name of the origin into the dialogue box and click the “OK” button. The origin is displayed in the world coordinate system.

Create Camera - Creates a new camera.

Operation

① Select Create Camera. The “Set Camera Name” dialogue box is displayed.

② Enter the name of a camera into the dialogue box and click the “OK” button. The camera is displayed at the origin of the world coordinate system.

Move Origin Only - Operates only the origin.

Short Cut: “Origin Only tool” in the tool palette

Only the origin can be operated, with the other objects protected against operation. Not displayed, the origin is displayed.

Operation

Select Move Origin Only.

Move Camera Only - Protects the objects other than the camera against operation.

Short Cut: “Camera tool” in the tool palette

Only the camera can be operated, with the other objects protected against operation. Not displayed, the camera is displayed.

Operation

Select Move Camera Only.

Link - Sets parenthood.

Supports the same function as the “Link tool” in the tool palette. Select the Link command instead of clicking the “Link tool”.

Unlink - Releases the set parenthood.

Supports the same function as the “Unlink tool” in the tool palette. Select the Unlink command instead of clicking the “Unlink tool”.

2.6 Window Menu

Cascade - Displays the window in a cascade.

Operation

Select Cascade.

Tile - Displays the window in a tile.

Operation

Select Tile.

The Top window is located at the top left. The General window is located at the top right. The Front window is located at the bottom left. The LeftSide window is located at the bottom right.

Arrange Icons - Arranges icons.

Arranges the displayed window icons.

Operation

Select Arrange Icons.

General - Displays the General window.

The displayed General window is given as an icon.

Operation

Select General.

Top - Displays the Top window.

The displayed Top window is given as an icon.

Operation

Select Top.

Front - Displays the Front window.

The displayed Front window is given as an icon.

Operation

Select Front.

Left Side - Displays the Left Side window.

The displayed Left Side window is given as an icon.

Operation

Select LeftSide.

2.7 Help Menu

About - Displays the version of the program currently used.

Checks the DLL version needed for the tool. If the version is not correct, the program is terminated.

Operation

Select About. The “About” dialogue box is displayed. To close the dialogue box, click the “About” dialogue box.

2.8 Tool Palette

Left Figure: Default mode

Center Figure: Mode where the axes operated by the mouse are restricted.

Right Figure: Mode for selecting the coordinate system operated through value input

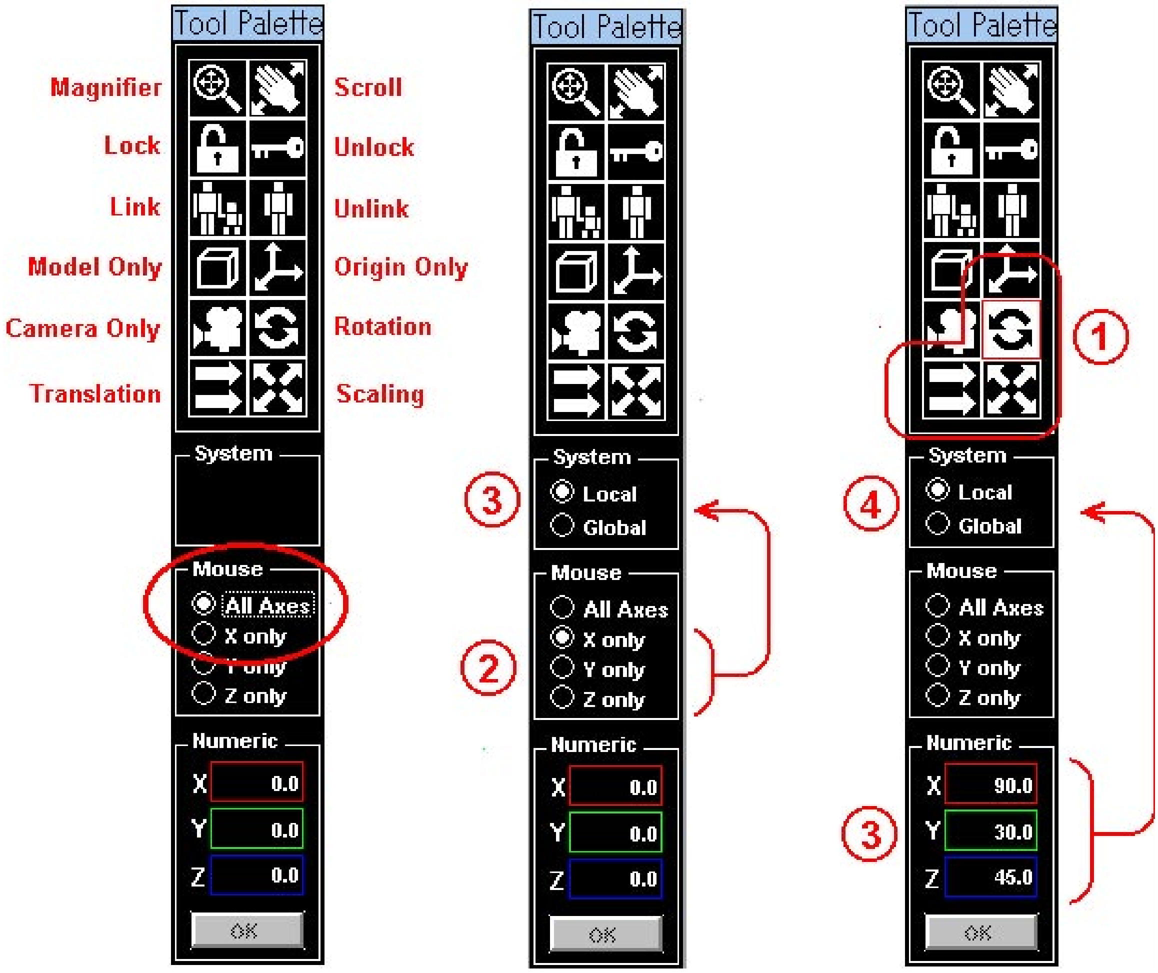


Figure 4-2-12 Tool Palette

Mouse - Restricts the axes to be operated.

Restricts the direction of parallel displacement by the mouse, or the axis of rotation. With axes restricted, the coordinate system as the reference of operation can be selected in the “System” column in the tool palette.

Operation (See the above center Figure.)

① Select an object.

② Select an axis to be operated in the “Mouse” column.
The “Local” and “Global” radio buttons are displayed in the “System” column in the tool palette.

All axes:	Not restricted.
X only:	Restricts the axis to be operated to the axis parallel to the X axis and passing through the center of the bounding box.
Y only:	Restricts the axis to be operated to the axis parallel to the Y axis and passing through the center of the bounding box.
Z only:	Restricts the axis to be operated to the axis parallel to the Z axis and passing through the center of the bounding box.

③ Select the coordinate system as the reference of operation in the “System” column in the tool palette.

④ Operate the object.

Note

This function is not associated with model extension and reduction.

With any of X, Y and Z Only selected, move the mouse to the top left or right on the screen for operation.

Numeric

Numeric input is used for operation. For numeric input, the coordinate system as the reference as operation can be selected in the “System” column on the tool palette.

Operation (See the above right Figure.)

① Select the tool for the intended operation from the tool palette.

Parallel displacement:	Translation tool
Rotation:	Rotation tool
Extension/reduction:	Scaling tool

② Select an object.

③ Enter parameters into the “Numeric” column. The “Local” and “Global” radio buttons are displayed in the “System” column on the tool palette.

④ Select the coordinate system as the reference of operation in the “System” column on the tool palette.

⑤ Press the return key or click the “OK” button.

Note

This function invalidates the setting of the mouse.

Remarks

The Tab key can be used for movement in the “Numeric” column.

System - Selects the coordinate system as the reference of operation.

Local: Coordinate system where the center of the bounding box is the origin and the axes are parallel to and in the same direction as the appropriate axes in the local coordinate system for the camera, the origin and a model after conversion

Global: Coordinate system where the center of the bounding box is the origin and the axes are parallel to and in the same direction as the appropriate axes in the World coordinate system

Remarks

Only for parallel displacement, with “All Axes” (see the above left Figure) selected in the “Mouse” column on the tool palette, another coordinate system is set. In the coordinate system, the center of the bounding box is the origin and the axes are parallel to and in the same direction as the appropriate axes in the visual point coordinate system for the camera for the window.

Magnifier tool - Magnifies and displays window contents.

Related commands: “Magnify” and “Minify” in the View menu

Magnifies or minifies by a factor of two, and displays a window.

Operation

◇ Magnification for display

① Click the “Magnifier tool”.

② Click the center of magnification.

◇ Minification for display

① Click the “Magnifier tool”.

② With the Shift key pressed, click the window to be minified.

Note

The window cannot be remodeled.

Scroll tool - Has the window scroll.

Operation

① Click the “Scroll tool”.

② Click the center of the window.

Mote

The window cannot be remodeled.

Lock tool - Locks parameters for the selected object.

Operation

① Click the “Lock tool”.

② Select an object to be locked. The locked object is displayed in the blue bounding box is protected against changes in the position, attitude and hierarchical structure by direct operation.

Note

With an object with an ancestor locked, the object is affected by operating the ancestor.

If the object having a created sequence has been locked, the sequence can be reproduced, but cannot be edited.

Unlock tool - Unlocks an object.

Operation

① Click the “Unlock tool”.

② Select an object to be unlocked.

Link tool - Creates hierarchical structure.

Related command: “Link” in the Tool menu

With object A set as the parent and object B set as a child, object B can be moved by moving object A. This is called “linking”. Parent selection leads to child selection. The parent and child objects can be moved and rotated as a unit. Further, only the child can be selected for movement and rotation. One object can be allocated more than one child.

Operation

① Select an object to be set as the child.

② Click the “Link tool”.

③ Select an object to be set as the parent. For the selected parent, the green bounding box is blinked. (For the selected child, the red bounding box is blinked.)
Setting another parent involves the selection of an object different from the selected parent. Continuous linking requires that the empty window be clicked to release the selected child. Then, repeat steps (1) to (3).

Note

A descendant can be selected as the parent.

Unlink tool - Releases hierarchical structure.

Related command: “Unlink” in the Tool menu

Operation

① Click the “Unlink tool”.

② Select the object set as a child. The bounding box for the selected parent is blinked in green.

Model Only tool - Allows only a model to be operated.

Operates only a model without moving the camera and the origin.

Operation

With this tool appearing in black on the white base as a result of clicking, only a model can be operated.

To release this mode, click this tool again.

Exception

The current camera can be operated by the keyboard and the camera can be operated by “Option” in the Set View dialogue box in this mode as well.

Origin Only tool - Allows only the origin to be operated.

Related commands: “Show Origin” in the View menu and “Move Origin Only” in the Tool menu

Only the model can be operated without moving any models and camera.

Operation

With this tool appearing in black on the white base as a result of clicking, only the origin can be operated.

To release this mode, click this tool again.

Exception

The current camera can be operated by the keyboard and the camera can be operated by “Option” in the Set View dialogue box in this mode as well.

Camera Only tool - Allows only the camera to be operated.

Related commands: “Show Camera” in the View menu and “MoveCameraOnly” in the Tool menu

Only the camera can be operated without moving any models and camera.

Operation

With this tool appearing in black on the white base as a result of clicking, only the camera can be operated.

To release this mode, click this tool again.

Translation tool - Allows only parallel displacement

Related commands: “Mouse”, “Numeric” and “System” on the tool palette

Operation

① Click “Translation tool”.

② Select an object to be subjected to parallel displacement.

③ Select a plane of the bounding box and drag it to the desired position. When moving the plane in the depth direction, drag it with the shift key pressed. Moving the mouse toward yourself causes the plane to near yourself. Moving the mouse away from yourself causes the plane to leave yourself.

For correct parallel displacement

① Click “Translation tool”.

All items in the “Numeric” column on the tool palette are reset to 0.0.

② Select an object to be subjected to parallel displacement.

③ Enter the amount of parallel displacement into the “Numeric column” on the tool palette.

④ Press the return key or click the “OK” button.

Note

Steps ① and ② can be reversed. This means that an object may have been selected.

Rotation tool - Allows only rotation.

Related commands: “Mouse”, “Numeric” and “System” on the tool palette

Operation

① Click “Rotation tool”.

② Select an object to be rotated.

③ Drag an edge or vertex in the bounding box for rotation to the desired position.

For correct rotation

① Click “Rotation tool”. All items in the “Numeric” column on the tool palette are reset to 0.0.

② Select an object to be rotated.

③ Enter into Numeric Control the degree of rotation from the current attitude about the X, Y and Z axes in the world coordinate system with respect to local coordinates. (Units = degrees)

④ Click the “OK” button.

Note

Steps (1) and (2) can be reversed. This means that an object may have been selected.

Scaling tool - Allows only extension or reduction.

Related commands: “Mouse”, “Numeric” and “System” on the tool palette

Operation

① Click “Scaling tool”.

② Select a model to be extended or reduced.

③ Select a vertex of the bounding box. Dragging the vertex away from the center of the bounding box provides extension. Dragging the vertex toward the center provides reduction.

Note

Selecting and operating a vertex displayed near the center results in a rapid size change. The following gives operation for accurate extension or reduction.

① Click “Scaling tool”. All items in the “Numeric” column on the tool palette are reset to 1.0.

② Select a model to be extended or reduced.

③ Enter into Numeric Control the power of extension or reduction along the X, Y and Z axes in the local coordinate system. (0 < power < 8)

④ Click the “OK” button.

Note

“Scaling tool” is applicable only to the models. The tool must not be applied to the origin and the camera.

Steps ① and ② can be reversed. This means that an object may have been selected.

2.9 Key Frame Recorder

Sequence creation requires that key frames as nodes of operation be required. Recording two or more key frames leads to automatic linear interpolation between the key frames and to sequence creation.

2.9.1 Names and Functions of Components

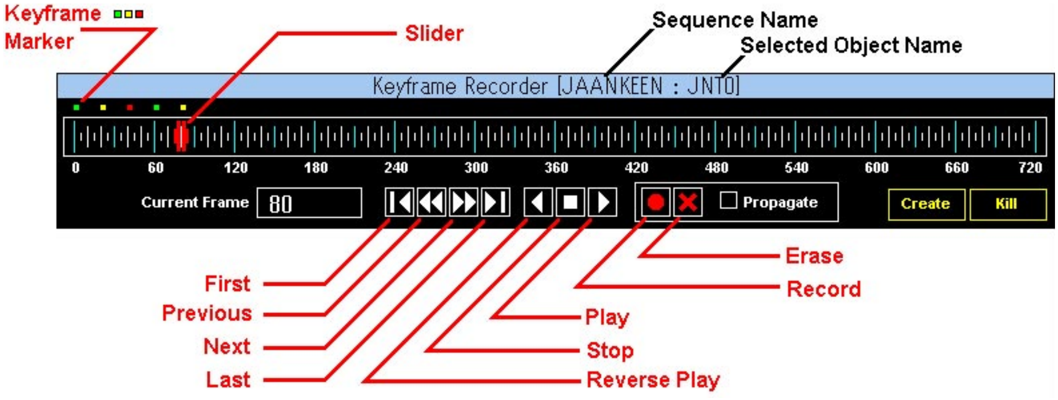


Figure 4-2-13 Key Frame Recorder

Keyframe Marker

The key frame marker indicates the position of the key frame marker set for the selected object. The following gives the meaning of colors.

Green: Signifies that the key frame contains an instruction for copying object data from PlayStation's CD-ROM into the RAM.
Yellow: Signifies that the key frame contains no instruction. (The marker is created in the frame following the green marker.)
Red: Signifies that the key frame contains an instruction for discarding an object from PlayStation's RAM.

The current frame can be changed to the key frame by clicking the key frame marker.

The key frame can be updated by dragging the key frame marker.

Note

The color of the key frame marker depends on the status of the Create and Kill buttons during key frame recording. (See the following Figure.)

With no object selected, the key frame marker is not displayed.

Create and Kill buttons

These buttons are used to set the flag having meaning only during animation execution by the PlayStation unit at the key frame.

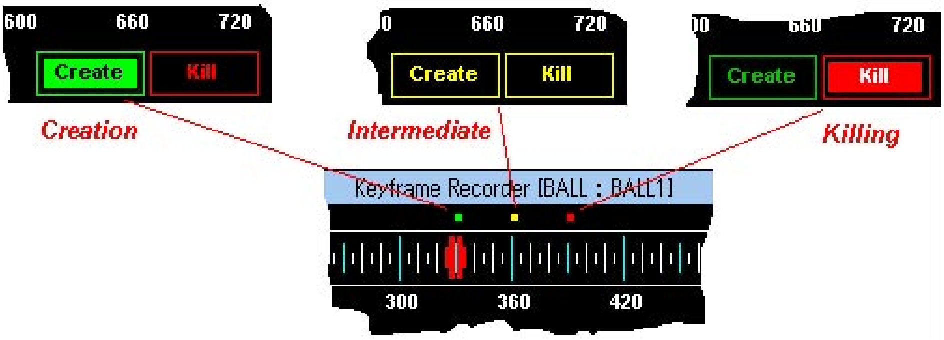


Figure 4-2-14 Create and Kill Buttons

Create button:

If the result of clicking this button is Creation (above left Figure), the green key frame marker is created. If the result is Intermediate (above center Figure), the yellow key frame marker is created.

Kill button:

If the result on clicking this button is Killing (above right Figure), the red key frame marker is created. If the result is Intermediate (above center Figure), the yellow key frame marker is created.

Note

On the PC, the object displayed on the screen is not changed, even though Create or Kill has been specified. For reproduction by the PlayStation unit, however, the object is created or discarded by this information. Thus, the object before creation and after discarding is not displayed on the TV set.

For the meaning of the color of the key frame marker, see paragraph for the key frame marker.

Slider

Indicates the current frame. Dragging the slider allows the current frame to be updated.

Graduation

Clicking the graduation at the current frame position enables the current frame to be updated.

CurrentFrame text box

Indicates the current frame. Pressing the return key with a numeric character entered in this column allows the setup of this value as the current frame. The values not less than zero and not more than 720 can be entered.

Current frame control button

Sets the key frame for the selected object as the current frame.

First button:	Start key frame
Previous button:	Key frame preceding the current frame
Next button:	Key frame following the current frame
Last button:	Last key frame

Note

This button is valid, only when the selected object has the key frame.

Animation button

Play button:	Plays animations from the current frame.
Reverse Play button:	Plays animations reversely from the current frame.
Stop button:	Stops the animation.

Note

This button is valid, only when the selected object has more than one key frame.

Propagate check box

Specifies whether to propagate the setting or deletion of the key frame to the descendants of the object as well.

Such setting and selection is propagated if the box has been checked.

Record button

Sets the key frame of the selected object. Modifying the key frame requires resetting by this button.

Note

The Record button is valid, only when an object is selected.

Erase button

Erases the created key frame. Click the marker of the key frame to be erased and, then, the “Erase” button.

Note

The Erase button is valid, only when an object is selected.

2.9.2 Setting Key Frame

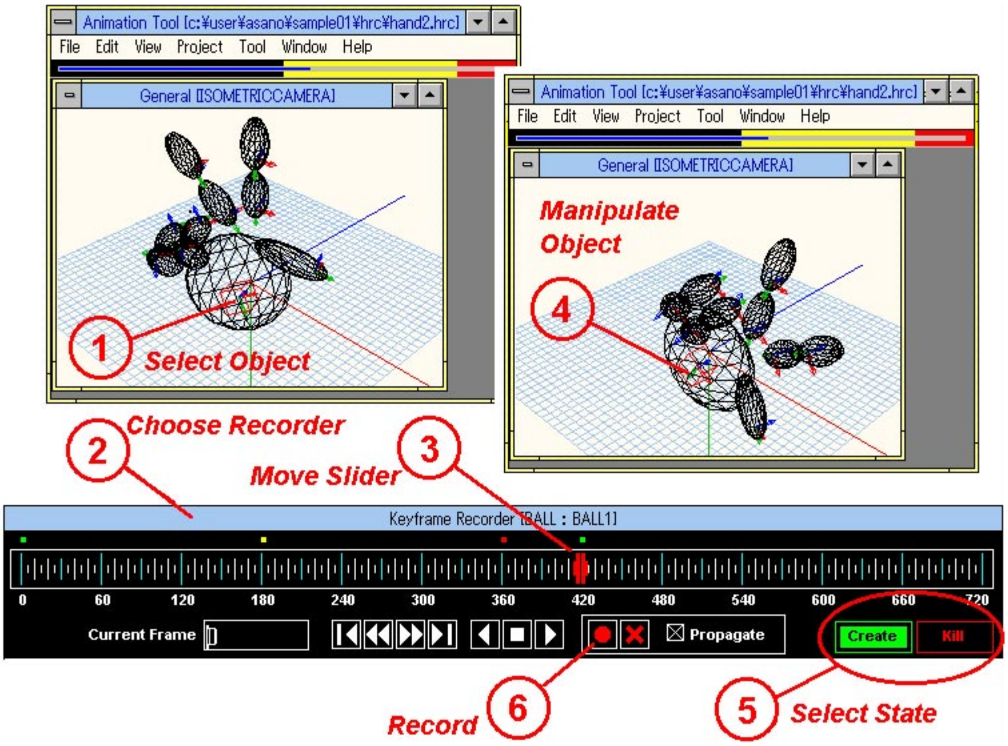


Figure 4-2-15 Setting Key Frame

Operation

① Select an object whose key frame is to be set.

② Select Keyframe Recorder. If the selected object is provided with the set key frame, the key frame marker is displayed.

③ Move the slider to the frame where the key frame is to be set. On the screen, the object corresponding to the frame is displayed.

④ Operate the object to determine its position and attitude. Before the key frame is set, this status is not recorded.

⑤ Click one of the “Create” and “Kill” buttons.

⑥ Click the “Record” button and set the key frame. Carry out steps (3), (4), (2), (5) and (6) in that order if needed. (If yellow has been set up, step (5) can be omitted.)

Note

No key frame is set if the Record button is not pressed. Carrying out any of the following operations after step (4) results in the loss of object location results without being recorded.

◇ The current frame is changed (except for dragging the slider as pressing the Ctrl key).

◇ An object different from the currently selected object is selected.

2.9.3 Key Frame Correction 1: Modification of object position and attitude, or Create or Kill

Operation

① Select an object.

② Select Keyframe Recorder.

③ Click the key frame marker to search for the key frame to be corrected.

④ Operate the object to update the position and attitude, and change the Create and/or Kill buttons.

⑤ Click the “Record” button.

2.9.4 Key Frame Correction 2: Copying of object position and attitude

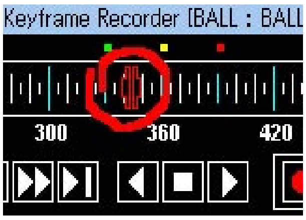


Figure 4-2-16 Copying of Object Position and Attitude

Operation

① Select an object.

② Select Keyframe Recorder.

③ To provide for copying, drag the slider to update the current frame.

④ With the Ctrl key pressed, drag the slider to the frame to be copied. The slider is changed as shown above.

⑤ Click the “Record” button. The slider display is returned to the previous status.

Note

The above displayed slider indicates that the status currently displayed in the window may not reflect interpolations on the frame.

2.9.4 Key Frame Correction 3: Movement of key frame marker

This method allows the modification of acceleration in the same order of key frames. Further, the key frame order can be changed.

Operation

① Select an object.

② Select Keyframe Recorder.

③ With the Ctrl key pressed, drag the key frame marker.

Important!

Upon the selection of the key frame marker, the mouse cursor points to the sand clock. After the cursor has been returned to the original state, start to drag the key frame marker.

2.10 Polygon Indicator

Figure 4-2-17 Polygon Indicator

Indicates the total number of polygons in a model. The above Figure indicates about 1400 polygons.

Yellow zone:	3,000 or more polygons: Beyond the value recommended for the PlayStation unit.
Red zone:	5,400 or more polygons: Old version left to provide for polygons in addition to the yellow zone

Note

Polygons for the camera and origin are not covered, as they are not passed to the PlayStation unit.

2.11 Limitations

1. Displaying the keyframe recorder involves screen resolution of 1024 x 768 dots or more.

2. Screen color customization is not supported. Some setting makes buttons, switched and icons invisible. On the basis of the comparison of the actual display with settings in this manual, the standard Windows settings should be introduced.

3. Any windows should be clicked only if required. Events of re-rendering are accumulated in the queue, and it takes longer time to carry out the subsequent operation. Too excessive clicking repetition results in an overflow of the event buffer, with an alarm produced. Ignoring the alarm results in system down.

4. The maximum number of polygons which can be loaded is about 20 times more than the recommended value (3,000 polygons) for the PlayStation unit. This value depends on the currently operated application and data.

5. The Scroll and Magnify/Minify commands are inapplicable to a remodeled window. Remodeling requires that this tool be turned off. The tool active during animation reproduction may result in screen trouble.

6. The setup of the previous screen with a window covered by other windows and dialogue boxes may leaves the displayed picture chipped. But the program and data are not affected. Carry out the following operation.

   To re-render all the four screens, press the F5 key. To re-render a certain screen, click the screen free of any objects.

7. To re-render only the moved object, the object position becomes white.

   The program and data are not affected. If more positions have become white, carry out the operation in item 6 to re-render the screen.

8. Upon the following operation, a ghost of the bounding box appears. But the program and data are not affected. To remove the ghost, carry out the operation in step 6 for re-rendering.

   Undoing, screen remodeling, or key frame setting in the order of object operation and slider operation

9. Before closing, the dialogue box for specifying whether to save the HRC file does not pop up.

CHAPTER 5

MISCELLANEOUS

1 ABOARD.EXE

1. Overview

ABOARD.EXE is an application for Windows for setting an I/O address to access the artist board (DHL-H201A). The I/O address set by this tool is looked into by the programs which use the artist board including the material editor, the sprite editor, the TIM utility, and the plug-in module for PhotoShop.

2. Operation

Upon the startup of ABOARD.EXE from the file manager, the following window pops up.

Figure 5-1-1 ABOARD.EXE Main Window

Upon the first startup of ABOARD.EXE, 0 x 1340 is displayed as the address. This is the factory set default for the artist board. If the I/O address has been changed by the dip switch on the board, use this application to modify the address setting.

An address can be set for each of the upper six bits and lower ten bits.

Saving the set value calls for pressing the “OK” button. Canceling the set value requires that “Cancel” be pressed. “Close” in the window menu is equivalent to “Cancel”. Pressing “OK” causes the set I/O address to be saved into $WINDOWS $ABOARD.INI.

3. Remarks

According to “address setting” in the instruction manual for the artist board, 0 x 134 is set in the factory. Note that the lower four bits are omitted. On the screen for this application, 0 x 1340 is displayed.

For the upper six bits, all values can be set.

For the lower ten bits, only the following values can be selected.

110, 120, 130, 140, 150, 180, 190, 1A0, 1B0, 1C0, 1D0, 1E0,

220, 230, 240, 250, 260, 280, 290, 2A0, 320, 330, 340, 350

If “OK” has been selected, the following warning dialogue box may pop up.

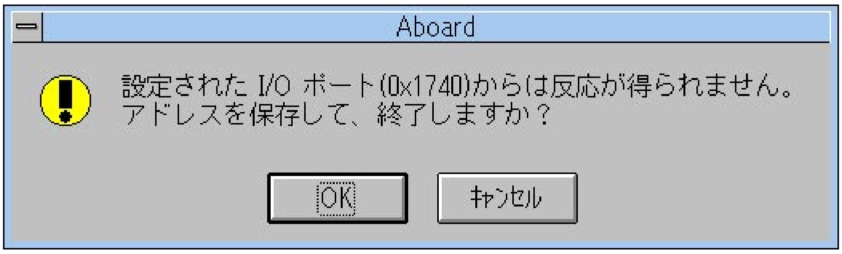


Figure 5-1.2 Warning Dialogue Box

This means that the artist board has not correctly reacted to the set I/O address.

Check artist board connection or address setting by the dip switch. If the specified address is used in any board other than the artist board, this warning dialogue box may pop up.

Selecting “OK” in the warning dialogue box causes the set address to be saves forcibly. Selecting “Cancel” causes the setting screen to appear.

Typically, in PC/AT or its compatible machine, only the lower ten bits are valid as the I/O address. For the artist board, however, all 16 bits are valid. For the I/O address set in this application, the lower 10 bits must be unique against the I/O address to be used in any board other than the artist board.

APPENDIX

APPENDIX

APPENDIX FILE FORMAT

This appendix explains the file format for the 3D graphics tool.

MODEL DATA

(RSD FORMAT)

1 Summary

The RSD data file format is a format for model data handled by an artist tool group. The RSD format is not for a single file. Four kinds of files are used to describe one model.

RSD file: Describes relationships between PLY/MAT/GRP and texture files.
PLY file: Describes positional information on vertices of a polygon.
MAT file: Describes material information on a polygon.
GRP file: Describes grouping information.

Information on combinations of the four files is stored in the RSD file. Thus, the files constituting objects can be accessed only by specifying the RSD file.

Because descriptive information is stored in more than one file, an object having a combination of different materials in a PLY file can be expressed.

Every file is a text file. Data is processed in units of lines delimited by a line feed character (LF or CR/LF). A comment line beginning with # can be inserted into any line.

Note: User of DTL-S230 Starter Kit

The present latest format is 940102. The format for Starter Kit (@RSD940101) is somewhat changed.

Thus, the documents and converter belonging to Starter Kit must be replaced by new ones.

Data prepared in the @RSD940101 format can be loaded by the material editor, but is converted into the new format when it is saved.

2 PLY File

The RSD file stores information on combinations of PLY, MAT and GRP files constituting a 3D object. A set of files is used to describe a single 3D object.

2.1 Whole Configuration

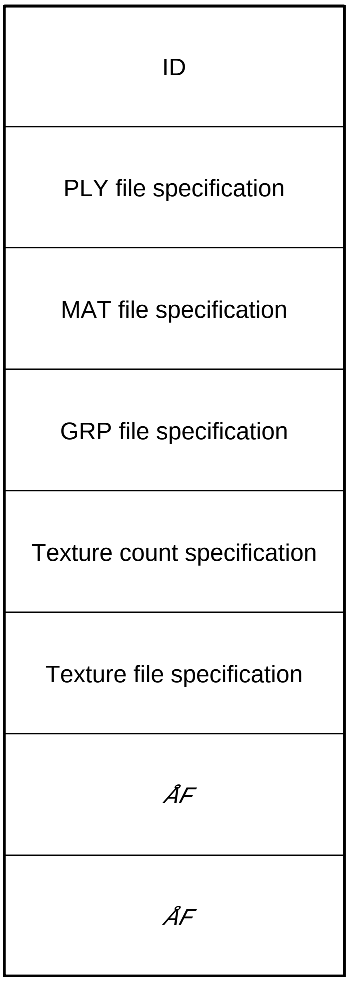


Figure 1 Whole RSD File Configuration

2.2 ID

The ID is composed of a character string that indicates the version of the RSD file format, being “@RSDnnnnnn” (where nnnn is a number). The current version is “@RSD940102”.

2.3 PLY File

PLY = (File name of PLY)

2.4 MAT File

MAT = (File name of MAT)

2.5 GRP File

GRP = (File name of GRP)

2.6 Texture Count

Specifies the number of textures used.

NTEX = (Number of textures)

2.7 Texture File

Specifies an image data file in the TIM format to be used as the texture, and the same number of texture files as a value specified by above NTEX. For the RSD file for a model which uses no texture, this block does not exist.

TEX[n] = (n-th texture file name)

File name description depends on the type of a processing system (MS-DOS, UNIX, etc.). Care should be taken for transfer between different kinds of machines.

2.8 Sample

The following gives a simple example of the RSD file.

```
@RSD940102
PLY=sample.ply
MAT=sample.mat
GRP=sample.grp
NTEX=3
TEX[0]=texture.tim
TEX[1]=texture2.tim
TEX[2]=texture3.tim
```

3 PLY File

The PLY file stores the positions of the vertices of polygons. The coordinate system for the PLY file is the same as for the extended library (libgs), with the X axis (forward) representing the right screen, the Y axis the bottom, and the Z axis the depth.

The direction (obverse or reverse) of a single-faced polygon is determined by the order of describing vertices in apolygon group. The plane for which the vertices of a polygon are described CW in the space is defined as the obverse of the polygon.

3.1 Whole Configuration

ID
Data length record
Vertex group
Normal group
Polygon group

Figure 2 Whole PLY File Configuration

3.2 ID

This is a character string representing the version of a PLY file format, being “@PLYnnnnnn” (where nnnn is a number). The present version is “@PLY940102”.

3.3 Data Length Record

Describes the number of data pieces (lines) for the subsequent five data blocks. Items are delimited by a blank or tab. (This also applies to the following elements.)

Number of vertices	Number of normals	Number of polygons

Figure 3 PLY File Data Length Record

3.4 Vertex Group

A vertex group is composed of three floating-point values representing coordinates of a vertex. One line serves one vertex.

x coordinate value	y coordinate value	z coordinate value

Figure 4 Vertex Descriptor for PLY File

3.5 Normal Group

A normal group is composed of three floating-point values representing the components of a normal vector.

x component	y component	z component

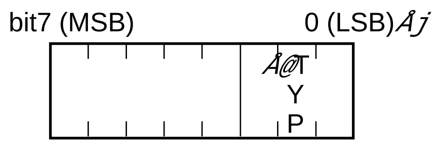
Figure 5 Normal Descriptor for PLY File

3.6 Polygon Group

A polygon group is composed of a flag for representing the type of a polygon, and nine parameters constituting the polygon. The meaning of the parameters varies with the value of a TYP field in the flag.

Flag	Parameter #1	Parameter #2	…	Parameter #9

(Flag bit configuration)



TYP: Defines the type of a polygon

000 : Triangle
001 : Quadrangle
010 : Straight line
011 : Sprite
1XX : reserved

Figure 6 PLY File Polygon Descriptor

[Polygon (Triangle or Quadrangle)]

The parameter section describes the vertices and normals for a polygon.

The values of vertices (0-3) and normals (0-3) are integers indicating the position of data in a data group. (The start is zero.)

For a polygon to be subjected to flat shading, the normal of each vertex has the same value, and the value of the first vertex is adopted. For a polygon to be subjected to smooth shading, the normal of each vertex has a different value.

The flag is a hexadecimal integral value representing the type of a polygon, and is not provided with a prefix of ‘0x’.

For a triangular polygon, the data (vertex 3 and normal 3) corresponding to the fourth vertex is assigned a value of zero.

For a quadrangular polygon, the vertices are described so that the first three vertices form a triangle and the second, third and fourth vertices form another triangle.

Flag	Vertex 0	Vertex 1	Vertex 2	Vertex 3	Normal 0	Normal 1	Normal 2	Normal 3

Figure 7 Polygon

[Straight line]

The parameter section describes the vertex numbers of two end points.

Flag	Vertex 0	Vertex 1	0	0	0	0	0	0

Figure 8 Straight Line

[Sprite]

A sprite in model data is rectangular image data located in a 3D space. It can be considered to be a textured polygon always facing the visual point.

The parameter section describes vertices indicating sprite positions, and the width and height of images (sprite patterns).

Flag	Vertex 0	WIDTH	HEIGHT	0	0	0	0	0

Figure 9 Sprite

3.7 Sample

The following gives a simple example of a PLY file.

```
@PLY940102
# Number of Items
8 12 12
# Vertex
0   0   0
0   0 100
0 100   0
0 100 100
100   0   0
100   0 100
100 100   0
100 100 100
# Normal
0.000000E+00 0.000000E+00 -1.000000E+00
0.000000E+00 0.000000E+00 -1.000000E+00
1.000000E+00 0.000000E+00 -0.000000E+00
1.000000E+00 0.000000E+00 0.000000E+00
0.000000E+00 0.000000E+00 1.000000E+00
0.000000E+00 0.000000E+00 1.000000E+00
-1.000000E+00 -0.000000E+00 -0.000000E+00
-1.000000E+00 0.000000E+00 0.000000E+00
-0.000000E+00 1.000000E+00 0.000000E+00
0.000000E+00 1.000000E+00 0.000000E+00
0.000000E+00 -1.000000E+00 0.000000E+00
0.000000E+00 -1.000000E+00 0.000000E+00
# Polygon
0 6 2 0 0 0 0 0 0
0 6 0 4 0 1 1 1 0
0 7 6 4 0 2 2 2 0
0 7 4 5 0 3 3 3 0
0 3 7 5 0 4 4 4 0
0 3 5 1 0 5 5 5 0
0 2 3 1 0 6 6 6 0
0 2 1 0 0 7 7 7 0
0 7 3 2 0 8 8 8 0
0 7 2 6 0 9 9 9 0
0 4 0 1 0 10 10 10 0
0 4 1 5 0 11 11 11 0
```

4 MAT File

4.1 Whole Configuration

ID
Number of materials
Material descriptor
:
:
:

Figure 10 Whole Configuration of MAT File

4.2 ID

This is a character string representing the version of a MAT file format, being "@MATnnnnnn" (where nnnn is a number). The present version is "@MAT940801".

New attributes of colored texture and gradation texture unavailable to the past format (@MAT940102) are supported in @MAT940801.

4.3 Number of Materials

Describes the number of subsequent material descriptors (lines).

4.4 Material Descriptor

Specifies a polygon and describes material information on the polygon.

Polygon No.	Flag	Shading	Material information

Figure 11 Material Descriptor

[Polygon number]

Indicates the number of a polygon in a polygon group in a PLY file. (The start is zero.) Range specification allows two or more polygons to be described in one line.

Description	Polygon of interest
1	1 only
0-5	0 1 2 3 4 5
2,4,6	2 4 6

[Flag]

This is a hexadecimal integer representing the type of a polygon. The flag is not provided with a prefix of ‘0x’. The following gives the meaning of each bit.

Bit 0: Light source calculation mode
- 0: Light source calculation supported
- 1: Fixed color

With light source calculation supported, the rendering color is determined by the directions of the light source and polygon. For the fixed color, the color is constant irrespective of the direction.

Bit 1: Back face
- 0: Single-faced polygon
- 1: Double-faced polygon

Bit 2: Flag on semitransparency
- 0: Not made to be semitransparent
- 1: Made to be semitransparent

With the flag set at 1, the polygon with no texture is always made to be semi-transparent, and the polygon with texture is made to be semitransparent/opaque/transparent depending on the STP bit of texture data.

Bits 3 to 5: Rate of semitransparency
- 000: 50% back + 50% polygon
- 001: 100% back + 100% polygon
- 010: 100% back – 100% polygon
- 011: 100% back + 25% polygon
- 1XX: reserved

The current library does not provide the capability to change the semitransparency rate of a polygon with no texture.

Bits 6 to 7: Reserved (Must be 0)

[Shading]

This is a character indicating the shading mode.

F: Flat shading   G: Smooth shading

With flat shading specified, the normal information specified in the PLY file for the first vertex is adopted.

[Material information]

The format is different depending on the material type (texture supported or not supported).

Texture not supported (Colored polygon/straight line)

TYPE	*R*	*G*	*B*

TYPE: Material type, whose value is “C”
R, G, B: RGB components of polygon color (0 to 255)

Figure 12 Texture not Supported (Colored Polygon/Straight Line)

Texture not supported (Gradation colored polygon/straight line)

TYPE	*R0*	*G0*	*B0*	*R1*	*G1*	*B1*	*ÅC□*	*R3*	*G3*	*B3*

TYPE: Material type, whose value is “G”
Rn, Gn, Bn: RGB components of the n-th vertex. For a triangular polygon, the RGB value of the fourth vertex is 0, 0, 0.

Figure 13 Texture not Supported (Gradation colored polygon/straight line)

Textured polygon/sprite

TYPE	TNO	U0	V0	U1	V1	U2	V2	U3	V3

TYPE: Material type, whose value is “T”
TNO: TIM data file to be used (Texture number described in the RSD file)
Un, Vn: Position of vertex n in the texture space. For a triangular polygon, the value (U3, V3) of the fourth vertex is zero.

Figure 14 Textured Polygon/Sprite

Colored textured polygon

TYPE	TNO	U0	V0	U1	V1	U2	V2	U3	V3	R	G	B

TYPE: Material type, whose value is “D”
TNO: TIM data file to be used. (Texture number described in the RSD file)
Un, Vn: Position of vertex n in the texture space. For a triangular polygon, the value (U3, V3) of the fourth vertex is zero.
R, G, B: RGB components of polygon color (0 to 255)

Figure 15 Colored Textured Polygon

* The colored textured polygon is used to make the texture of a polygon bright without light source calculation. This type allows the three-dimensional drawing of a textured object without light source calculation. It is valid only in the fixed color light source calculation mode.

Gradation textured polygon

TYPE	*TNO*	*U0*	*V0*	*U1*	*V1*	*U2*	*V2*	*U3*	*V3*

R0	*G0*	*B0*	*R0*	*G1*	*B1*	*Åc□*	*R3*	*G3*	*B3*

TYPE: Material type, whose value is “H”
TNO: TIM data file to be used. (Texture number described in the RSD file)
Un, Vn: Position of vertex n in the texture space. For a triangular polygon, the value (U3, V3) of the fourth vertex is zero.
Rn, Gn, Bn: RGB components of the n-th vertex (N = 0 to 3). For a triangular polygon, the RGB value of the fourth vertex is 0, 0, 0.

Figure 16 Gradation Textured Polygon

* The gradation textured polygon is used to provide the same effect as textured smooth shading without light source calculation. This type is valid only in the fixed color light source calculation mode.

4.5 Sample

The following gives a simple example of the MAT file.

```
@MAT940801
# Number of Items
10
# Materials
0-5  0 F C 255 255 255
6    0 G T 1 10 0 25 71 40 25 0 0
7    0 G T 1 10 30 20 75 40 25 0 0
8    0 G T 1 18 73 30 79 40 25 0 0
9    0 G T 1 12 23 29 77 40 25 0 0
10   0 F T 1 18 13 75 72 40 25 0 0
11   0 F T 0 22 10 24 74 40 25 0 0
12   0 F T 0 30 39 41 79 40 25 0 0
13   1 F D 0 116 47 118 77 69 46 69 77  30 187 187
14   1 F H 0 69 46 69 77 17 45 15 77  101 210 138 52 211 188 101 210
```

5 GRP File

A group of polygons in the PLY file can be assigned a name.

Thus, a group of polygons can be operated by the material editor, and certain polygons can be accessed from the program.

5.1 Whole Configuration

ID
Number of groups
Group descriptor
⋮

Figure 17 Whole Configuration of GRP File

5.2 ID

This is a character string representing the version of a GRP file, being "@GRPnnnnnn" (where nnnn is a number). The current version is "@GRP940102".

5.3 Number of Groups

Covers the number of subsequent group descriptors.

5.4 Group Descriptor

Defines the configuration of a group. A group descriptor is composed of two or more lines.

Start line

Group name	Polygon No. line count	Number of polygons

Group name: Name assigned to a group
Polygon No. line count: Number of subsequent lines for polygon No. description
Number of polygons: Number of polygons belonging to a group

Figure 18 GRP Descriptor (Start line)

[Subsequent line (for polygon No. description)]

Specifies the numbers of polygons belonging to a group. The value indicates the position of a polygon in the PLY file. Range specification allows two or more polygons to be described in one line.

Description	Polygon of interest
1	1 only
3-7	3 4 5 6 7
2,4,6	2 4 6

5.5 Sample

The following gives a simple example of the GRP file.

```
@GRP940102
# Number of Groups
2
# Group list
 upper_part 2 5
10-13
25
lower_part 3 3
3
5
7
```

MODEL DATA FOR LIBRARY (TMD FORMAT)

1 Outline

TIM format data is model data handled by 3D library libgs.

The data can be migrated onto the memory and used as arguments for libgs functions as it is.

In the phase of such artist tools as the 3D graphics tool, the RSD file storing more abstract text data is used, with the data converted into this format by RSDLINK.EXE during program creation.

TIM data is a group of such primitives as object component polygons and straight lines. One TMD file can serve more than one object.

2 About Coordinate Values

Coordinate values in the TMD file follows space handled by the 3D library. The positive direction of the x axis represents the right, the y axis the bottom, and the z axis the depth. The spatial coordinate value of each object is a signed integral value of 16 bits, ranging from -32768 to +32767.

In the phase of design, or in the RSD format, the vertex value is a floating-point value. Thus, conversion from RSD into TMD involves extension or reduction for scale adjustment. The adjusted value is reflected in the object structure described later as the reference value.

The value provides an index for mapping with respect to world coordinates. Current libgs ignores the scale value.

3 Format

The TMD file stores a table of 3D objects, and three kinds of data of primitives, vertices and normals. It is composed of four blocks.

HEADER
OJB TABLE :
PRIMITIVIE :
VERTEX :
NORMAL :

Figure 1 Whole File Structure

4 HEADER

The header section is composed of three word (12 bytes) data carrying information on data structure.

ID
FLAGS
NOBJ

ID: Data having 32 bits (one word). Indicates the version of a TMD file. The current version is 0x00000041.

FLAGS: Data having 32 bits (one word). Carries information on TIM data configuration. The least significant bit is FIXP. The other bits are reserved and their values are all zero. The FIXP bit indicates whether the pointer value of the OBJECT structure described later is a real address. A value of one means a real address. A value of zero indicates the offset from the start.

NOBJ: Integral value indicating the number of objects

Figure 2 Structure of Header

5 OBJ TABLE

The OBJ TABLE section contains a set of structures having pointer information on where objects are stored.

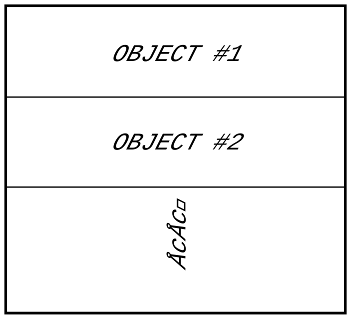


Figure 3 Structure of OBJ TABLE

One object structure has the following configuration.

```
struct object {
      u_long *vert_top;
      u_long n_vert;
      u_long *normal top;
      u_long n_normal;
      u_long *primitive top;
      u_long n_primitive;
      long scale;
}
```

(Explanation of members)

vert_top:	Start address of a vertex
n_vert:	Number of vertices
normal_top:	Start address of a normal
n_normal:	Number of normals
primitive_top:	Start address of a primitive
n_primitive:	Number of primitives

Figure 4 Object Structure

Among the structure members, the pointer value has meaning depending on the value of the FIXP bit in the header section. A FIXP value of 1 indicates a real address, while a FIXP value of 0 indicates a relative address with respect to the start of the object section as an address of zero.

The scaling factor is of a signed long type, and the value raised to the second power indicates the scale value.

A structure scale value of zero provides a full-scale value.

A value of two provides a scale value of four. A value of -1 provides a scale value of 1/2. These values can be used to set up the scale used for design.

6 PRIMITIVE

The primitive section is composed of a set of object primitive rendering packets. One packet refers to one primitive.

The primitive defined by TMD is different from the rendering primitive handled by libgpu.

A function supported by libgs assumes the tasks of perspective transformation and conversion into the rendering primitive.

The packet is of variable length, and its size and structure vary with the primitive type.

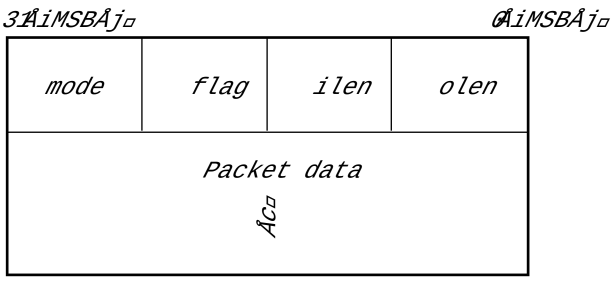
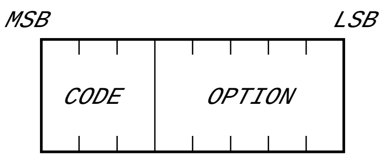


Figure 5 Typical Structure of Rendering Packet

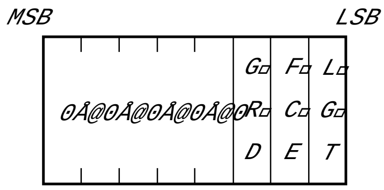
Mode:



CODE: 3-bit code representing the type of a component
001 = Polygon (Triangle/quadrangle)
010 = Straight line
011 = Sprite (Rectangle)
OPTION: Varies with the values of options, bit and CODE.
(See a list of packet data configurations described later.)

Figure 6 Mode

Flag:



GRD: Valid only for the polygon not textured, subjected to light source calculation
- 1: Gradation polygon
- 0: Single-color polygon

FCE:
- 1: Double-faced polygon
- 0: Single-faced polygon

(Valid, only when the CODE value refers to a polygon.)

LGT:
- 1: Light source calculation not carried out
- 0: Light source calculation carried out

Figure 7 Flag

ilen: Word length (8 bits) of packet data section

olen: Word length (8 bits) of rendering primitive created by intermediate processing

Packet data: Parameters for vertices and normals

Packet data varies with the primitive type.

7 VERTEX

The vertex section is composed of a set of structures representing vertices. The following gives the format of one structure.

MSB	LSB
VY	VX
-	VZ

VX, VY, XZ: x, y and z values of vertex coordinates (16-bit integer)

Figure 8 Vertex Structure

8 NORMAL

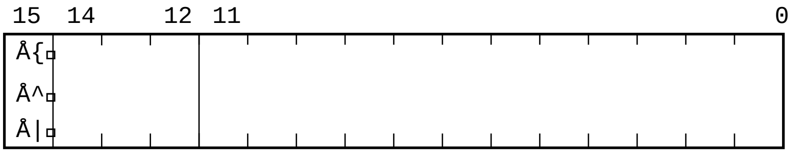
The normal section is composed of a set of structures representing normals. The following gives the format of one structure.

MSB	*LSB*
NY	*NX*
-	*NZ*

NX, NY, NZ: x, y and z components of a normal (16-bit fixed-point value)

Figure 9 Normal Structure

Each of NX, NY and NZ values is a signed 16-bit fixed-point value where a number of 4096 is considered to be 1.0.



Sign: 1 bit
Integral part: 3 bits
Decimal part: 12 bits

Figure 10 Fixed-Point Format

9 Packet Data Composition Table

This section lists packet data configurations for each primitive type.

The following are the parameters contained in the packet data section:

Vertex(n): Index value of 16-bit length pointing to a vertex. Indicates the position of the element from the start of the vertex section for an object covering the polygon.

Normal(n): Index value of 16-bit length pointing to a normal. Same as Vertex.

Un, Vn: x and y coordinate values on the texture source space for each vertex

Rn, Gn, Bn: RGB value representing polygon color being an unsigned 8-bit integer. Without light source calculation, the predetermined brightness value must be entered.

TSB: Carries information on a texture/sprite pattern.

bit 15 … 8	7 6	6 5	4 … bit 0
0 0 0 0 0 0 0	TPF	ABR	TPAGE

TPAGE : Texture page number (0 to 31)

ABR : Semitransparency rate (Mixture rate).

Valid, only when ABE is 1.

00	50%back + 50%polygon
01	100%back + 100%polygon
10	100%back - 100%polygon
11	100%back + 25%polygon

TPF : Color mode

00	4 bit
01	8 bit
10	15 bit

Figure 11 TSB

CBA: Indicates the position where CLUT is stored in the VRAM.

bit15	14 … 6	5 … bit 0
	CLY	CLX

CLX: Upper six bits of 10 bits of X coordinate value for CLUT on the VRAM

CLY: Nine bits of Y coordinate value for CLUT on the VRAM

Figure 12 CBA

10 Triangular Polygon: With Light Source Calculation

Bit configuration of mode value

MSB LSB

0	0	1	IIP	0	TME	ABE	TGE

IIP: Shading mode
- 0: Flat shading
- 1: Gouraud shading

TME: Texture specification
- 0: OFF
- 1: On

ABE: Semitransparency processing
- 0: OFF
- 1: ON

TGE: Brightness calculation during texture mapping
- 0: ON
- 1: OFF (Texture is rendered as it is.)

(This also applies to the other polygons.)

Figure 13 Mode Value for Triangular Polygon and Light Source Calculation Supported

Packet configuration

Flat, No-Texture

0x20	0x00	0x03	0x04
0x20(Note)	B	G	R
Vertex0		Normal0	
Vertex2		Vertex1	

Gouraud, No-Texture

0x30	0x00	0x04	0x06
0x30(Note)	B	G	R
Vertex0		Normal0	
Vertex1		Normal1	
Vertex2		Normal2	

Flat, No-Texture (Gradation)

0x20	0x04	0x05	0x06
0x20(Note)	B0	G0	R0
Å[	B1	G1	R1
Å[	B2	G2	R2
Vertex0		Normal0	
Vertex2		Vertex1	

Gouraud, No-Texture (Gradation)

0x30	0x04	0x06	0x06
0x30(Note)	B0	G0	R0
Å[	B1	G1	R1
Å[	B2	G2	R2
Vertex0		Normal0	
Vertex1		Normal1	
Vertex2		Normal2	

Flat, Texture

0x24	0x00	0x05	0x07
CBA		V0	U0
TSB		V1	U1
Å[	Å[	V2	U2
Vertex0		Normal0	
Vertex2		Vertex1	

Gouraud, Texture

0x34	0x00	0x06	0x09
CBA		V0	U0
TSB		V1	U1
Å[	Å[	V2	U2
Vertex0		Normal0	
Vertex1		Normal1	
Vertex2		Normal2	

Note: Same value as mode

In the above example, the values of mode and flag indicate a single-faced polygon and semitransparency processing not carried out

Figure 14 Packet Configuration for Triangular Polygon and Light Source Calculation Supported

APPENDIX

11 Quadrangular Polygon: With Light Source Calculation

Bit configuration of move value

MSB … LSB

0	0	1	IIP	1	TME	ABE	TGE

(For each bit value, see “Triangular Polygon and Light Source Calculation Supported”.)

Figure 15 Mode Value for “Quadrangular Polygon and Light Source Calculation Supported”

Packet configuration

Flat, No-Texture

0x28	0x00	0x04	0x05
0x28(Note)	B	G	R
Vertex0		Normal0	
Vertex2		Vertex1	
Å\		Vertex3	

Gouraud, No-Texture

0x38	0x00	0x05	0x08
0x38(Note)	B	G	R
Vertex0		Normal0	
Vertex1		Normal1	
Vertex2		Normal2	
Vertex3		Normal3	

Flat, No-Texture (Gradation)

0x28	0x04	0x07	0x08
0x28(Note)	B0	G0	R0
Å[	B1	G1	R1
Å[	B2	G2	R2
Å[	B3	G3	R3
Vertex0		Normal0	
Vertex2		Vertex1	
Å[		Vertex3	

Gouraud, No-Texture (Gradation)

0x38	0x04	0x08	0x08
0x38(Note)	B0	G0	R0
Å[	B1	G1	R1
Å[	B2	G2	R2
Å[	B3	G3	R3
Vertex0		Normal0	
Vertex1		Normal1	
Vertex2		Normal2	
Vertex3		Normal3	

Flat, Texture

0x2c	0x00	0x07	0x09
CBA		V0	U0
TSB		V1	U1
Å[	Å[	V2	U2
Å[	Å[	V3	U3
Vertex0		Normal0	
Vertex2		Vertex1	
Å[		Vertex3	

Gouraud, Texture

0x3c	0x00	0x08	0x0c
CBA		V0	U0
TSB		V1	U1
Å[	Å[	V2	U2
Å[	Å[	V3	U3
Vertex0		Normal0	
Vertex1		Normal1	
Vertex2		Normal2	
Vertex3		Normal3	

Note: Same value as mode

Figure 16 Packet Configuration for "Quadrangular Polygon and Light Source Calculation Supported"

12 Triangular Polygon: Without Light Source Calculation

Bit configuration of mode value

MSB LSB

0Å@	0Å@	1Å@	IIP	Å@0	TME	ABE	TGE

(For each bit value, see “Triangular Polygon and Light Source Calculation Supported”.)

Figure 17 Mode value for “Triangular Polygon and Light Source Calculation not Supported”

Packet configuration

Flat, No-Texture

0x21	0x01	0x03	0x04
0x21(Note)	B	G	R
Vertex1		Vertex0	
Å[		Vertex2	

Gouraud, No-Texture

0x31	0x01	0x05	0x06
0x31(Note)	B0	G0	R0
Å[	B1	G1	R1
Å[	B2	G2	R2
Vertex1		Vertex0	
Å|		Vertex2	

Flat, Texture

0x25	0x01	0x06	0x07
CBA		V0	U0
TSB		V1	U1
Å[	Å[	V2	U2
Å[	B	G	R
Vertex1		Vertex0	
Å[		Vertex2	

Gouraud, Texture

0x35	0x01	0x08	0x09
CBA		V0	U0
TSB		V1	U1
Å[	Å[	V2	U2
Å[	B0	G0	R0
Å[	B1	G1	R1
Å[	B2	G2	R2
Vertex1		Vertex0	
Å|		Vertex2	

Note: Same value as mode #2.

Figure 18 Packet Configuration for “Triangular Polygon and Light Source Calculation not Supported”

13 Quadrangular Polygon: Without Light Source Calculation

Bit configuration of mode value

MSB LSB

0Å@0Å@1Å@	I P	Å@1	TME	ABE	TGE

(For each bit value, see “Triangular Polygon and Light Source Calculation Supported”.)

Figure 19 Mode Value for “Quadrangular Polygon and Light Source Calculation not Supported”

Packet configuration

Flat, No-Texture

0x29	0x01	0x03	0x05
0x29(Note)	B	G	R
Vertex1		Vertex0	
Vertex3		Vertex2	

Gouraud, No-Texture

0x39	0x01	0x06	0x08
0x39(Note)	B0	G0	R0
Å[	B1	G1	R1
Å[	B2	G2	R2
Å[	B3	G3	R3
Vertex1		Vertex0	
Vertex3		Vertex2	

Flat, Texture

0x2d	0x01	0x07	0x09
CBA		V0	U0
TSB		V1	U1
Å[	Å[	V2	U2
Å[	Å[	V3	U3
Å[	B	G	R
Vertex1		Vertex0	
Vertex3		Vertex2	

Gouraud(Gradation), Texture

0x3d	0x01	0x0a	0x0c
CBA		V0	U0
TSB		V1	U1
Å[	Å[	V2	U2
Å[	Å[	V3	U3
Å[	B0	G0	R0
Å[	B1	G1	R1
Å[	B2	G2	R2
Å[	B3	G3	R3
Vertex1		Vertex0	
Vertex3		Vertex2	

Note: Same value as mode

Figure 20 Packet Configuration for “Quadrangular Polygon

APPENDIX

and Light Source Calculation not Supported”

14 Straight Line

Bit configuration of mode value

MSB							LSB
0	1	0	IIP	0	0	ABE	0

IIP: Gradation
1: Gradation ON
0: Gradation OFF (Single color)
ABE: Semitransparency processing
1: ON
0: OFF

Figure 21 Mode Value for “Straight Line”

Packet configuration

Gradation OFF

0x40	0x01	0x02	0x03
0x40(Note)	B	G	R
Vertex1		Vertex0	

Gradation ON

0x50	0x01	0x03	0x04
0x50(Note)	B0	G0	R0
Å l	B1	G1	R1
Vertex1		Vertex0	

Note: Same value as mode

Figure 22 Packet Configuration for “Straight Line”

MODEL DATA FOR LIBRARY (PMD FORMAT)

15 3D Sprite

This sprite has a 3D coordinate value, and the contents of rendering are the same as for the typical sprite.

Bit configuration of mode value

MSB … LSB

0	1	1	SIZ	1	ABE	0

SIZ: Sprite size
00Å@Free size (Specified by W and H)
01Å@1Å~1
10Å@8Å~8
11Å@16Å~16

ABE: Semitransparency processing
1ÅFON
0ÅFOFF

Figure 23 Mode Value for “3D Sprite”

Packet configuration

Free size

0x64	0x01	0x03	0x05
TSB		Vertex0	
CBA		V0	U0
H		W	

1Å~1

0x6C	0x01	0x02	0x04
TSB		Vertex0	
CBA		V0	U0

8Å~8

0x74	0x01	0x02	0x04
TSB		Vertex0	
CBA		V0	U0

16Å~16

0x7c	0x01	0x02	0x04
TSB		Vertex0	
CBA		V0	U0

Figure 24 Packet Configuration of “3D Sprite”

1 Outline

PMD format data is a kind of model data handled by extended graphics library libgs in the high-speed processing format. Compared with the TMD format, some functions are simplified, enabling high-speed processing.

The following are the objects handled by PMD data:

- Triangular and quadrangular polygons
- Data including packet creation areas
- A set of polygons with the same attribute

2 About Coordinate Values

A coordinate value in a PMD file follows the space handled by the 3D library, with the positive direction of the x axis representing the right, the y axis the bottom and the z axis the depth.

The spatial coordinate value of an object is a signed 16-bit integral value ranging -32768 to +32767.

3 Format

The PMD file contains a 3D object table of primitives and vertices.

ID:	Data of 32-bit length (one word) indicating the version of a PMD file. The current version is 0x00000042.
PRIM POINT:	32-bit integral value (in bytes) indicating the offset of the PRIMITIVE Gp section from the start of a file
VERT POINT:	32-bit integral value (in bytes) indicating the offset of the VERTEX GP section from the start of a file. For an independent vertex, this value is zero.
OBJ TABLE:	Object array
PRIMITIVE Gp:	A set of polygon groups with the same attribute
VERTEX Gp:	A row of vertex coordinates for a shared vertex

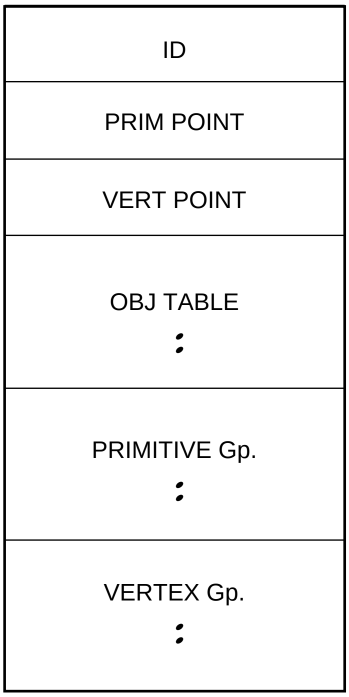
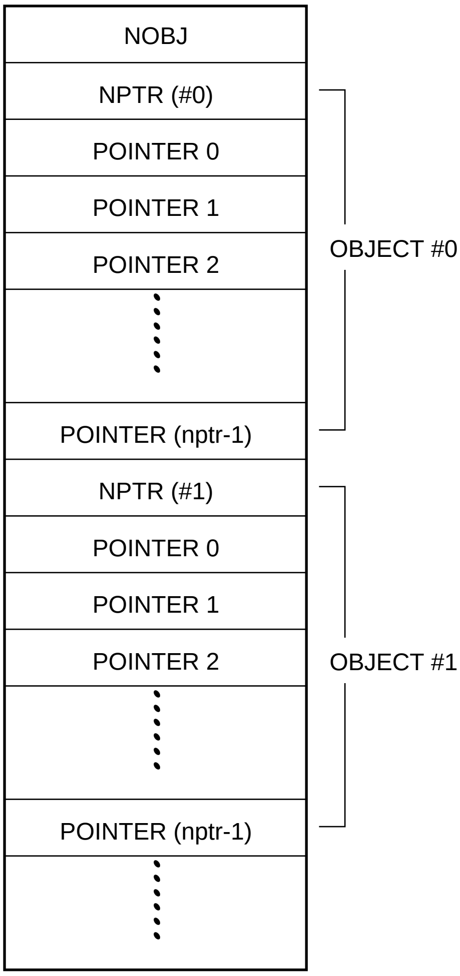


Figure 1 Whole File Structure

4 OBJ TABLE

The OBJ TABLE section is a table of pointer information on primitive groups constituting objects.

A single object is composed of primitive groups.



NOBJ: Number of objects in OBJ TABLE
NPTR: Number of pointers in a single object
POINTER: Pointer to a primitive group

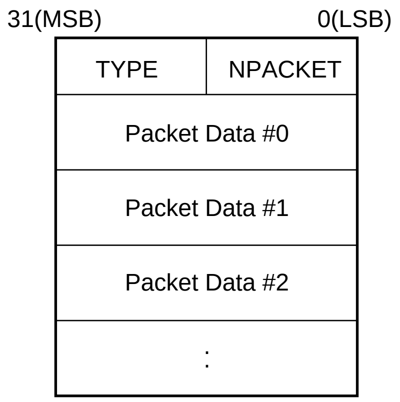
Figure 2 OBJECT Structure

5 PRIMITIVE Group

The primitive group section is composed of a set of rendering packets as object primitives. One packet refers to one primitive.

The primitive defined by PMD is different from the rendering primitive handled by libgpu. Functions supported by libgs assume the tasks of perspective conversion and conversion into a rendering primitive.

The following gives the structure of one primitive group.



NPACKET: Number of packets
TYPE: Packet type

Figure 3 Structure of PACKET Group

Bit configuration of TYPE

bit No.	0	1
16	3 Triangle	4 Quadrangle
17	Flat	Gouraud
18	Texture-On	Texture-Off

19	Independent vertex	Shared vertex
20	Light source calculation not supported	Light source calculation supported
21	Back clip provided	Back clip not provided
22~31	(Reserved)	

The configuration of packet data varies with the TYPE value.
The following gives a packet data configuration for each type.

Note 1: Two sets of primitive sections (POLY ***) for a structure are supported to provide for a double buffer. The contents of the two sets must have been initialized.

Note 2: Bits 20 and 21 do not affect the configuration of packet data.

TYPE=00 (Triangle/Flat/Texture-On/Independent vertex)

```
struct _poly_ft3 {
   POLY_FT3 pkt[2];
   SVECTOR v1, v2, v3;
}
```

TYPE=01 (Quadrangle/Flat/Texture-On/Independent vertex)

```
struct _poly_ft4 {
  POLY_FT4 pkt[2];
  SVECTOR v1, v2, v3, v4;
}
```

TYPE=02 (Triangle/Gouraud/Texture-On/Independent vertex)

```
struct _poly_gt3 {
   POLY_GT3 pkt[2];
   SVECTOR v1, v2, v3;
}
```

TYPE=03 (Quadrangle/Gouraud/Texture-On/Independent vertex)

```
struct _poly_gt4 {
   POLY_GT4 pkt[2];
   SVECTOR v1, v2, v3, v4;
}
```

TYPE=04 (Triangle/Flat/Texture-Of/Independent vertex)

```
struct _poly_f3 {
   POLY_F3 pkt[2];
   SVECTOR v1, v2, v3;
}
```

APPENDIX

TYPE=05 (Quadrangle/Flat/Texture-Off/Independent vertex)

```
struct _poly_f4 {
   POLY_F4 pkt[2];
   SVECTOR v1, v2, v3, v4;
}
```

TYPE=06 (Triangle/Gouraud/Texture-Off/Independent vertex)

```
struct _poly_g3 {
   POLY_G3 pkt[2];
   SVECTOR v1, v2, v3;
}
```

TYPE=07 (Quadrangle/Gouraud/Texture-Off/Independent vertex)

```
struct _poly_g4 {
   POLY_G4 pkt[2];
   SVECTOR v1, v2, v3, v4;
}
```

TYPE=08 (Triangle/Flat/Texture-On/Shared vertex)

```
struct _poly_ft3c {
```

IMAGE DATA

(TIM FORMAT)

```
        POLY_FT3 pkt[2];
        long vp1, vp2, vp3;
    }
```

TYPE=09 (Quadrangle/Flat/Texture-On/Shared vertex)

```
    struct _poly_ft4c {
        POLY_FT4 pkt[2];
        long vp1, vp2, vp3, vp4;
    }
```

TYPE=0a (Triangle/Gouraud/Texture-On/Shared vertex)

```
    struct _poly_gt3c {
        POLY_GT3 pkt[2];
        long vp1, vp2, vp3;
    }
```

TYPE=0b (Quadrangle/Gouraud/Texture-On/Shared vertex)
```
struct _poly_gt4c {
   POLY_GT4 pkt[2];
   long vp1, vp2, vp3, vp4;
}
```

TYPE=0c (Triangle/Flat/Texture-Off/Shared vertex)
```
struct _poly_f3c {
   POLY_F3 pkt[2];
   long vp1, vp2, vp3;
}
```

TYPE=0d (Quadrangle/Flat/Texture-Off/Shared vertex)
```
struct _poly_f4c {
   POLY_F4 pkt[2];
   long vp1, vp2, vp3, vp4;
}
```

TYPE=0e (Triangle/Gouraud/Texture-Off/Shared vertex)
```
struct _poly_g3c {
   POLY_G3 pkt[2];
   long vp1, vp2, vp3;
}
```

TYPE=0f (Quadrangle/Gouraud/Texture-Off/Shared vertex)
```
struct _poly_g4c {
```

```
    POLY_G4 pkt[2];
    long vp1, vp2, vp3, vp4;
}
```

pkt[ ] indicates the corresponding rendering primitive packet.

v1 to v4 indicates coordinate values of vertices.

vp1 to vp4 indicate offsets from the start of a row of shared vertices.

6 VERTEX

The vertex section is composed of a set of sector structures representing shared vertices. The following gives the format of one structure.

MSB LSB

VY	VX
-	VZ

VX, VY, XZ: x, y and z values (16-bit integers) of vertex coordinates

Figure 4 Vertex Structure

1 Outline

The TIM file can be transferred directly to the VRAM in the image standard.
It can be used as sprite patterns, as well as 3D texture mapping materials.
The following are the image data modes (color count)

A. 4bit CLUT
B. 8bit CLUT
C. 15bit Direct color
D. 24bit Direct color

The VRAM is of a 16-bit configuration. Hence, only 16-bit and 24-bit modes are supported to transfer data directly to the frame buffer for display. When data is used as sprite patterns or polygon texture mapping data, any of 4-bit/8-bit/16-bit modes can be selected.

2 File Configuration

The TIM file has a file header at its start, being composed of several blocks.

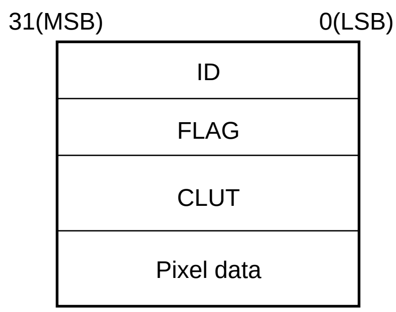


Figure Whole TIM File Configuration

Data is of the 32-bit binary format. Because of Little Endian, multi-byte data is arranged in ascending order. (See Figure 2.)

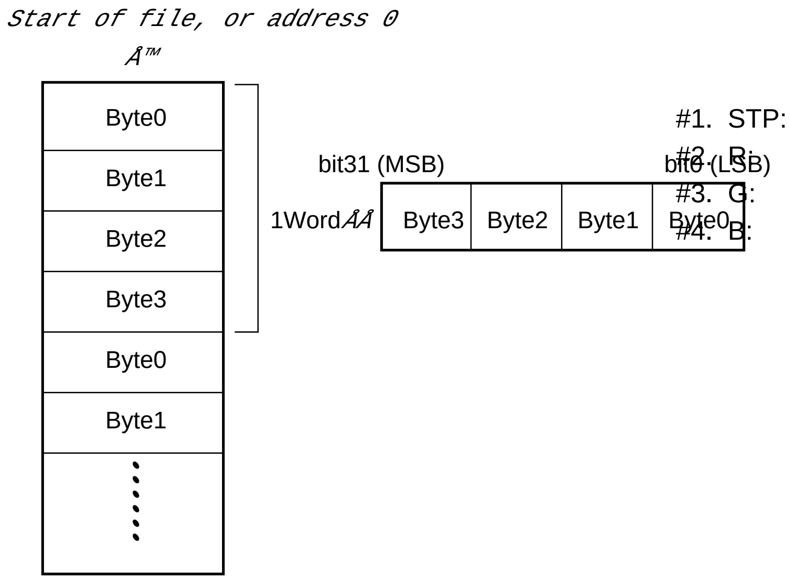
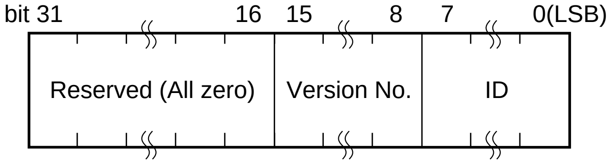


Figure 2 Byte Order in File 3. ID

3 ID

The file ID is a one-word data, having the following bit configuration.



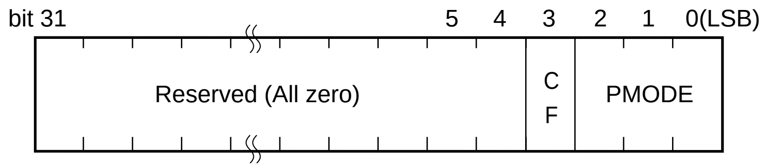
bit0~7 = ID: 0x10
bit8~15 = VERSION: 0x00

Figure 3 Structure of TIM File Header

APPENDIX

4 FLAG

The flag is data composed of 32 bits, providing information on data structure. The following gives its bit configuration.



bit0~2 = PMODE Pixel mode (Bit length)
- 0: 4bit CLUT
- 1: 8bit CLUT
- 2: 15bit Direct
- 3: 24bit Direct
- Others: reserved

bit 3 = CF Whether CLUT exists
- 0: There is no CLUT section
- 1: There is a CLUT section

Figure 4 Flag Bit Configuration

5 CLUT Part

Image data in the 4-bit/8-bit color mode uses CLUT.

If the value of a CF flag in the header section is ‘1’, the TIM file has CLUT (color palette).

The CLUT section begins with a CLUT byte count, followed by inner-VRAM positional information, image size, and the main data body.

bit31(MSB)	bit0(LSB)
bnum	
DY	DX
H	W
CLUT1	CLUT0
⋮	
CLUT n	CLUT n-1

bnum	Data length of CLUT section (in bytes) (Including 4 bytes of bnum)
DX	X coordinate in the VRAM
DY	Y coordinate in the VRAM
H	Longitudinal size of data
W	Lateral size of data
CLUT 0~n	CLUT entry (16 bits/entry)

Figure 5 Structure of CLUT Section

One CLUT set is composed of 16 CLUT entries for the 4-bit mode, and of 256 CLUT entries for the 8-bit mode.

CLUT is located on the VRAM. A CLUT section in a TIM file is handled as a rectangular VRAM image. This means that one CLUT entry is equivalent to one pixel in the VRAM.

Hence, one CLUT set is handled as rectangular image data of height = 1 and width = 16 for the 4-bit mode, and of height = 1 and width = 256 for the 8-bit mode.

INDEX

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

INDEX

P

Q

R

S

T

U

INDEX

V

Z

INDEX

INDEX

3D Graphics Tool User's Manual for 3D Graphics Tool
Version 2.0 PlayStation Artist Tool

PlayStation Artist Tool

Designed and prepared by Sony Computer Entertainment Inc.

Akasaka Ohji Building, 8-1-22, Akasaka, Minato-Ku, Tokyo 107
Phone: 03-3475-7300 (Key number)
For further information, please contact the business department on 03-3475-7320.