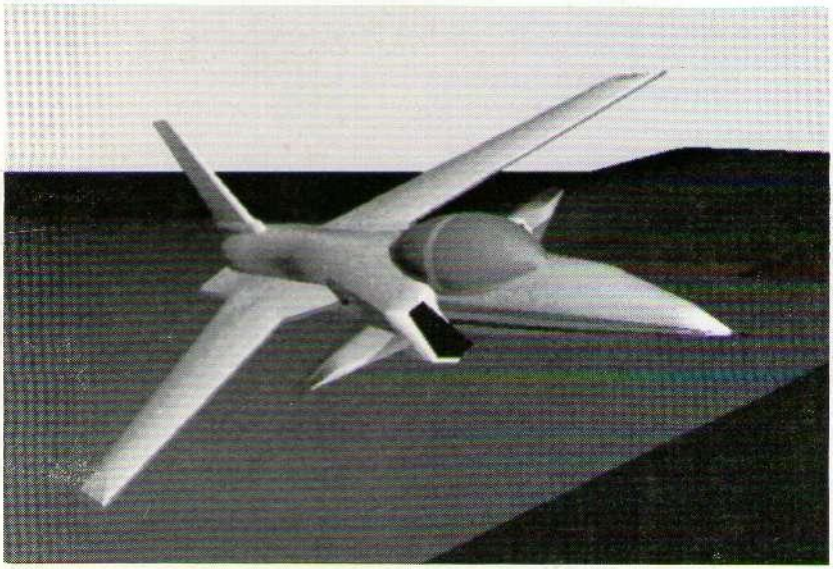
SCULPT-ANIMATE 4D Jr. *User Manual*



BYTE by BYTE™
CORPORATION


BYTE by BYTE™
CORPORATION

1294

SAJr. Serial Number - _____

PREFACE

Several years ago one of my professors gave an interesting definition of an animator.

"When you find someone who's good at animation," he said, "you will also have found an extremely weird bird."

His point was that animation required infinite patience, good math skills, artistic talent and technical savvy. Those of us who lacked these attributes were put on notice to turn our efforts to the more generalized techniques of film and video production. And leave animation to the *weird birds* of the world.

I don't know that Dr. Eric Graham ever heard this definition. Thanks to his Sculpt-Animate series of programs, however, anyone can produce an animation these days. Weird bird or not.

The Sculpt-Animate series is one of the most amazing graphics packages to reach the personal computer. The menus and methods are straightforward. Logical. Beneath this interface, then, lurks powerful 3D code--capable of operations that were once in the exclusive domain of mainframes.

Given this easy interface, someone who has some knowledge of 3D representations and/or animation might proceed to animating simple objects shortly after boot-up. For the rest of us, a few observations may be in order.

As you read this manual, look at the objects around you. Don't just see what they look like, but how they are shaped. You can see that almost every object has clear and definite features, outlines, and dimensions. Every feature has a certain position with respect to the rest of the object, and the world. These are the elements used to model objects in Sculpt-Animate.

After you are comfortable creating object models, sit back for an hour or so and watch a few animations. Observe how the characters really move. Study how limbs operate, how objects rotate and change shape. See how your eye blends many discrete images together, perceiving smooth motion. These are the elements used to create animations in Sculpt-Animate.

Read this manual once to acquaint yourself with the general method. This first reading is a nice time to link the Sculpt-Animate strategy with your own observations regarding objects and animation. By the second reading, you'll be like a kid on Christmas morning. As you do the tutorials, you'll begin to get the feeling that Dr. Graham has given you a *weird bird* all your very own. Your objects will be made to fly, bank, walk, and squash. In so doing, Sculpt-Animate will open up an exceedingly powerful medium limited only by the boundaries of your imagination.

Ten years hence, when rendering is done in real time, disk storage is tabulated in gigabytes, and the average RAM-count is the size of a Buick, Sculpt-Animate will still be around, lighting the way. Given these anticipated hardware advances, the Sculpt-Animate sequences of the future may be indistinguishable from camera-reality. And that brings us to another definition. My old college professor also reminded us that it doesn't take genius to make the complex... complex. "It takes genius to make the complex simple." By that definition, the Sculpt-Animate series is a work of genius. Sculpt-Animate flies fast and high. It also flies low enough so that non-computer/non-math types like me, guys that can't even spell *cowkulus*, much less do it, can jump onboard and do the animations we've always dreamed of.

Floyd Wray
Earth & Sky Films

FOREWORD

This manual and the software described in it are copyrighted with all rights reserved. Under the copyright laws, neither this manual nor the software may be copied, in whole or in part, without written consent of Byte by Byte Corporation. The software program may be copied for backup purposes, or installed on a single hard disk, or both. Multiple installations are expressly forbidden. Site license agreements are available.

In other words, you may (and should) make backup copies of the Sculpt-Animate 4D Jr. distribution disks. You may also copy the program onto a hard disk for more efficient use. Please do not make copies of the program for others. You are honoring the copyright if only one copy of the software you purchased is being used at any one time.

When we released Sculpt-Animate 4D Jr.'s predecessors, we took a risk. We thought low-cost software could be sold without copy protection. We were wrong. It seems that some people who never see what goes into software development don't understand that a couple of pounds of paper and plastic can be a bargain at $150. Yet this software compares with packages costing 100 times as much. Given the popular (extreme) falsehood that software developers are all rich, it is entirely too easy to give copies to friends, etc.

This has made continued development of Sculpt-Animate extremely difficult. It has literally become a decision of whether to kill the series or to implement some protection against illegal copying. We will not kill the series! Instead, we have chosen a form of protection--the simplest and least hazardous we could find.

It is this simple: you may run any copy of the program, but when it starts you will be asked to type in a word from a specific place in the manual. If you do this correctly, the program will proceed normally. This scheme is as painless (for all concerned) and as effective as any other.

Limited Warranty on Media and Manual:

If you discover physical defects in the media on which this software is distributed, or in the manual distributed with the software, Byte by Byte Corporation will replace the media or manual at no charge to you, provided you return the item to be replaced with proof of purchase to Byte by Byte Corporation within 90 days of the purchase date. Please include your warranty registration card if you have not already sent it in.

Registered Sculpt-Animate 4D Jr. owners may purchase a technical support policy at $15 for ten minutes or $75 per hour. Technical support is free of charge for 30 days after date of purchase. We **must** have your warranty registration card before any assistance can be given.

(512) 343-4357 (10am-5pm, CST)

Please do not call for technical assistance until you are sure that this manual does not answer your question. The manual is very complete, and it can explain the program much more effectively than we can over the phone. It will also be easier for you to describe your difficulty if you are familiar with the manual's explanation.

Copyright (C) 1989 Byte by Byte Corporation
Arboretum Plaza II
9442 Capital of Texas Highway North, Suite 150
Austin, TX 78759

SCULPT-ANIMATE 4D Jr. written by Dr. Eric Graham

Manual: Written by Cathryn E. F. Graham
Edited by Brian McCullough and Kelly Ford

Cover: Art by Marcus Brooks
Layout by Tony Tomsu, Scott Peterson, & Steve Bittick

Printing: AlphaGraphics #151 and The Whitly Company

Revision 1

CONTENTS

CHAPTER 1: AN OVERVIEW

SAJr. Objects 1-1
The SAJr. Universe 1-3
Images 1-4
Animations 1-5

CHAPTER 2: GETTING STARTED

The Menu 2-7
Loading An Image 2-9
Loading A Scene 2-13
Making A Picture 2-21
Saving An Image 2-24
Saving A Scene 2-24
Leaving SAJr. 2-25
If You Have Problems 2-25

CHAPTER 3: THE OBSERVER

Imaging Modes 3-1
Placing The Observer 3-3
Aiming The Observer 3-4
Camera Lenses 3-5
Exposure Control 3-6
Image Size On The Screen 3-7
Tilting The Camera 3-8
Starting The Image Making Process 3-9
Hiding And Restoring The Image 3-10
Aborting The Image Making Process 3-10
Memory Usage 3-11
Other Status Display Items 3-13

CHAPTER 4: BUILDING OBJECTS

Vertices 4-1
Edges 4-2
Faces 4-4
Selecting Vertices 4-5
Erasing Objects 4-10
Color 4-12
The Grabber 4-15
Techniques For Using The Grabber 4-17

CONTENTS

CHAPTER 5: POWER TOOLS

Spheres 5-1
Subdividing Objects 5-3
Other Curved Objects 5-3
Cubes 5-4
Prisms 5-5
Drawing Curves 5-6
The Edge Maker Tool 5-7
Spinning New Objects 5-8
Extruding Shapes 5-10

CHAPTER 6: MORE TOOLS

Saving And Loading Objects 6-1
Rotating Objects 6-3
Expansion And Contraction 6-5
Manipulating Shapes 6-7
Duplicating Objects 6-8
Building Mirror Images 6-8
Multiple Use Of The Extrude Tool 6-9
Creating More Complicated Objects 6-10

CHAPTER 7: THE WORLD

Creating Lamps 7-1
Illumination 7-3
The Sky 7-5
The Ground 7-6

CHAPTER 8: CREATING AN ANIMATION

The Take 8-1
The Global Scene 8-4
Key Frame Animation 8-5
Betweening 8-7
Betweening Distortion 8-11
An Example Of Animation 8-12
Previewing An Animation 8-14
Looking At Individual Frames 8-16
RAM Animation 8-17
Making A RAM Animation 8-19
Playing A RAM Animation 8-22

CHAPTER 9: ADVANCED ANIMATION TECHNIQUES
Loop Mode 9-1
Frame Duration 9-2
File Names Conventions 9-3
Disk Space 9-4
Organizing Floppy Disks 9-5
Saving Images 9-7
Color And Exposure Locking 9-8
Animating The Observer 9-9
Animating Lamps 9-10
Model Animation 9-10
Conclusion 9-11

APPENDIX A: FUNCTIONAL REFERENCE
The Geometry Of SAJr A-1
The Tri-View Gadgets A-3
Tri-View Mouse Usage A-6
Menu Commands A-7
The File Requester A-24
The Expansion Requester A-25
The Color Requester A-25
The Modify Faces Requester A-26
The Modify Lamps Requester A-26
The Modify Take Requester A-26

APPENDIX B: GLOSSARY

INDEX

Notes

INTRODUCTION

What Is Sculpt-Animate 4D Jr.?

Sculpt-Animate 4D Jr., (or SAJr. as we will abbreviate it in this manual), is an integrated solid modeling, animation, and image rendering system. It is the latest member of the Sculpt-Animate series. The program enables you to design three-dimensional (3D) objects and display them as still images, in full color. Objects can be made to move or change their shape, so animated images can also be created. The ability to handle changes with respect to time adds a fourth dimension to this powerful program.

This program differs from others in the Sculpt Animate series, in that it is aimed at those individuals that wish to acquire solid modeling and three dimensional animation skills with the minimum of effort.

You can use SAJr. to construct a computer simulation of virtually any 3D shape that your mind can conceive. Then you can see what it would look like by displaying its image on the computer screen. You don't have to be a mathematician or know anything about 3D geometry to use SAJr.

Any shape can be formed and manipulated with the aid of the mouse and its two buttons. You are not limited to simple geometric shapes like spheres and cylinders. Indeed, with a little imagination you can build 3D models of almost any real-world object. With some artistic skill, you can even create very difficult irregular shapes, such as human and animal forms.

Regard SAJr. as the next step beyond a paint program. A paint program enables you to daub simulated pigments on the screen of your display without having to deal with the physical encumbrances of painting, and without messing up

your best shirt. Using a paint program effectively requires artistic skill and particularly if you are trying to produce a naturalistic scene.

SAJr. takes care of many of the messy technical details that any 2D artist would have to deal with, such as perspective and illumination. In fact, making an image with SAJr. is very much like taking a photograph, just point and shoot and out comes a picture.

When you are using SAJr. to build a solid object, the construction is much easier than with any real construction medium. Changes can be made easily, and you can save your intermediate stages. If you make a mistake or change your mind you can quickly backtrack.

SAJr. is not limited to making still images. The data for some moving images can be compressed enough so that the animation can be displayed directly by the Amiga, this technique is called RAM-Animation and was used for the 'Juggler' and 'Kahnankas' demonstrations, as well as others that have been shown throughout the Amiga community.

Software packages like SAJr. have been available for some time. What makes SAJr. unique, besides its extra-simple interface, is that the powerful three dimensional object modeling and animation packages have traditionally cost about one hundred times as much and required hardware that cost a hundred times more than an Amiga.

What is the catch? Well, SAJr. can only display an image within the constraints of the Amiga hardware. It can never generate a big-screen image as detailed as those seen in the movie 'Tron'. Similarly, because the Amiga is not a supercomputer, some tasks that SAJr. can perform will take a while. Does your Amiga really work 24 hours a day? Now it can be doing useful work while you are asleep, or away at the office!

Even within its hardware-imposed constraints, SAJr. offers you a powerful ability--the ability to build your own private universe. Once you have finished construction, you

can roam your imaginary world and take pictures from any viewpoint, in any direction. Look at ordinary objects in extraordinary ways, impossible in the real universe, or look at objects that have never existed, and see what they'd look like if they did.

Indeed, while we have been emphasizing SAJr. as a recreational and artistic tool, it is by no means limited to this role. Professionals involved in construction or manufacturing-- of buildings, vehicles, coffee mugs, or whatever-- can use SAJr. to view designs before committing to actual production. Scientists can present or visualize objects that couldn't be seen otherwise, such as molecular shapes or mathematical constructs. Videographers can create flying logos and moving illustrations to give their work visual impact.

No special skills are needed to operate SAJr. You can 'take a picture' at any time to see if your 3-D objects are shown the way you'd like them to be. This manual will lead you through all the stages of solid modeling and image generation. Unlike many programs, you can start getting useful results from SAJr. with a minimum of training. In a few pages you will be told how to generate your first picture, and by the time you have finished the book, there will be virtually no scene that you cannot create.

If you wish to progress beyond the scope of SAJr., you can upgrade to Sculpt Animate 4D which has additional features for producing images of almost photographic realism. It even includes the effects of shadows and reflections! SA4D has advanced tools to aid the 3D modeler in creating incredibly complex objects. Designed for the future, SA4D supports graphics modes beyond the Amiga's current display capability. Images can be created with the Amiga in SA4D and displayed on high end machines. SA4D also enables you use sophisticated animation techniques such as object cycling and hierarchical motion control.

About This Manual

The main portion of this manual is organized as an introductory text. This text walks you through SAJr. feature by feature, explaining what each feature does and how it may be used in your creation of objects, images and animations.

Chapter 1 provides a grand overview of the program to whet your appetite and to give you a feeling for where you are and where you are going when you move on to the chapters that follow.

Chapter 2 introduces you to the use of solid modeling and static image generating capabilities in SAJr., and lets you familiarize yourself with the essential elements of operating the program. A section at the end of this chapter provides troubleshooting hints.

Chapters 3 through 7 thoroughly explore the static imaging and modeling aspects of the program, progressing from the basics through advanced techniques, and laying a foundation for animation.

Chapters 8 and 9 present the techniques of key frame animation which is a straightforward method of producing moving images.

A reference section follows the main text. This contains appendices which may be consulted for concise definitions of SAJr. functions and terminology.

Finally, an Index is provided after the reference section.

CHAPTER 1

AN OVERVIEW

This chapter contains an overview of SAJr.. Everything in this chapter is described elsewhere, so you may skip ahead if you wish. You may find it useful to read what follows, though, so that you have a grasp of the big picture before we delve into all the details of the program in the other chapters.

While SAJr. is easy to use, some aspects of its use are not immediately obvious, even if you are used to other Amiga programs. You may want to begin using the program immediately. This is a good way to get your feet wet, so to speak. Be careful, you can quickly get in over your head. If this happens, don't be too discouraged. Just turn back to this manual and continue reading.

You will find that there are several different ways to do certain things. If what you are trying to do seems too difficult, you can probably find an easier way if you review the related portions of this manual. When a reference is made to another chapter, be sure to look there as well.

SAJr. Objects

The objects that we will be building, whether they are cubes, aircraft or elephants, will all be made from a single, basic building block--a triangular face. Such faces can be joined together to approximate the surface of any shape. The smaller and more numerous the triangles, the better the approximation. It is up you to decide upon the degree of subdivision you use to represent a smooth surface.

Unless you plan to generate only wire frame images, each triangular face must be given a color. You are not limited to a palette of 16, 32, or even 4096 colors. Any face

can have any color. When you generate an image of a scene made up of many faces, the colors on the screen will be determined both by the color of a particular face and by the way it is illuminated. Finally, the restrictions of the Amiga display hardware must be imposed, but this will be done for you automatically.

Although the triangular face is the basic building block for objects and is the only visible object in the final, full color images, each face is described in terms of simpler components. These components, while not visible in the final image, define the size and position of the face and are called vertices and edges. A vertex describes a position in space and usually represents the corner of a face, while an edge is a link between two vertices and is used as the side of a face. Vertices and edges can be shared between more than one face, so SAJr. objects are really built up from a triangular network of edges. By moving one or more vertices, you can change the shape of an object.

Objects are built in an interactive environment called the Tri-View that displays the vertices and edges in a so-called wire frame view. By definition, wire frame displays are only capable of showing the edges of surfaces such as lines or points. The Tri-View displays wire frame objects but also shows additional details. The Tri-View lets you look at a box-like region of space from three different angles at once, usually northward, westward and downward. You can use the mouse to manipulate vertices in all three directions, but you will have to switch from one view to another to be sure that you have created the intended change.

You need to use multiple views because a window on a flat screen can only represent two dimensions at once. Think of a shoebox: if you look directly at its end, you can only see two of its dimensions--height and width. If you want to look at its length, you have to look at either its top or side. The Tri-View allows you to do this. Although it may seem difficult at first, especially if you are new to such things. Take heart, after a little practice, you will be surprised by the intricate structures you can build.

SAJr. gives you commands for producing many standard shapes automatically. These can then be stretched or distorted to produce still more shapes. Shapes that have an element of symmetry can often be generated from simple curves. For example, a wine bottle shape can be generated by building its profile and using the Spin command to sweep the profile around, thereby creating a curved surface. Similarly, the Extrude Tool will let you stamp out shapes like a cookie cutter.

Objects can be saved and recalled from disk, so you can quickly assemble a library of useful objects. Once built, these objects can be reused in any number of projects. Because of the success of the programs in the Sculpt Animate series, there are numerous objects that have been placed in the public domain by their builders, or included in commercially available object libraries. When you use pre-constructed objects, it is an almost trivial task to assemble interesting and complicated scenarios. Dazzling images can be created with ease.

The SAJr. Universe

Once you have assembled your objects, SAJr. can be asked to provide certain environmental elements to round out your world. This unique arrangement of objects, their placement, and the environment is referred to as a 'Scene'. Many elements have been provided to allow you to make your Scene as interesting as possible.

One useful item is a ground plane that stretches off to infinity. The ground can be a color, just like a face. A sky can also be specified. If your demands for a background setting are more complicated--for example if you need a distant mountain range, or clouds in the sky--then these can be created in the form of objects.

A major purpose of SAJr. is to create images of your modeled objects. What you see depends upon the direction in which you are looking as well as your location. Think of the Observer as a camera that can be positioned and pointed

anywhere. In the SAJr. universe, the 'Observer' remembers both orientation and position. The view that this hypothetical camera would capture is what your screen displays when an image is rendered. You are provided with a number of controls over the Observer, including a lens with a alterable focal length, so that you can zoom in or out of the scene and modify the apparent perspective of the image.

In addition to objects and the Observer, a scene will normally contain lamps to illuminate the objects. As any painter, theatrical set designer, or photographer knows, the placement of light sources can be instrumental in creating a desired effect. In SAJr., any number of lamps can be used, and each can be given a different position, color, and brightness.

Images

There are numerous ways to produce a computer generated image. As you might expect, the better quality images usually require more computer time. SAJr. is capable of producing two basic kinds of image.

A wire frame image displays only the edges of surfaces. These edges are drawn as simple lines. The wire frame images are similar to the view provided by the Tri-View, except that it is in the correct perspective for the observer's viewpoint. Wire frame images are useful for visualizing an object quickly, for the time required to generate the image is small. They are also useful as an end product when you wish to create a 'computerized' look.

A much more realistic image can be generated by displaying each face using a shade of color that depends upon the combination of the intrinsic color of that face and the light sources that are illuminating the scene.

Even more realistic ways of creating an image are provided by other members of the Sculpt Animate line of products. All the skills that you acquire using SAJr. can be

transferred to these sophisticated rendering programs, should you choose to upgrade at some future time. Objects and scenes created in SAJr. can be easily imported into these programs as well. Then you will be able to produce images with an almost photographic element of realism.

Images can be stored on disk in the Amiga's standard IFF format. Images files can also be recalled from disk to be displayed on the screen. They can be modified using either paint programs or image processing programs, although not every program can handle the overscan size that SAJr. is capable of generating.

Animations

SAJr. can be used to make animations. Objects can be made to move by specifying their positions at different times. Each feature of the SAJr. universe can be made to change. Lamps can dim or brighten and change color. Objects can change size and shape, or even vanish. The observer can fly through the scene and change its viewing direction. The sky can be made to darken to simulate dusk. There is practically no limit to the effects that can be animated.

An animation consists of many images, each differing from the previous one by just the right amount to create the illusion of smooth motion. SAJr. uses a mechanism called a 'Take' to keep track of all the details of a single animated sequence. The Take will keep track of the names of all the Scene files that describe the objects as they move around. It also keeps track of the name of each Image file that is generated. When a Take has been properly specified, the process of rendering all the frames can be accomplished with a single mouse click.

Object movement can be specified in two ways. You could, for instance, create a series of scenes by setting the position of each object for every picture in the animation. This would be extremely tedious. The second method is more convenient. Since most motion is smooth, it's much

easier just to set up a few Scenes called Key Frames. A Key Frame Scene would be created at key spots in the movement. The program can usually figure out where the objects should be for the frames in between. This technique is called Key Frame animation.

The motions can represent actions that are impossible in the real world. Objects can be made to tumble and gyrate, change their shapes and even pass through one another. It is up to you to decide what effect you wish to create. In the world of computer generated animations there are few limits to what can be achieved in the way of special effects.

If the animation is not too large, it is possible to store its images in memory in compressed form and then display them quickly enough to simulate smooth motion. At video frame rates, however, even a second's worth of animation can use a lot of memory. If the motion is extreme, the time required to reconstruct each image may prevent full speed playback. Nevertheless, RAM animations can be useful for demonstrations, and several RAM animations can be joined together onto video tape for longer presentations.

As you can see, SAJr. offers a great many facilities which you can use to create sophisticated images and animations. Sophistication is not a prerequisite for the user. While it is true that advanced tools and 'power user' techniques exist, the essence of SAJr. is to open up the world of 3D graphics and animation to everyone. Here it is, explore!

Notes

Notes

CHAPTER 2

GETTING STARTED

This chapter introduces you to the basics of operating SAJr. You will be shown how to load a scene from a disk, generate a picture of the scene, and finally save the picture to disk.

Before you do anything else, read the support policy statement and warranty information that came with this package. Fill out the warranty registration form and send it in. This ensures that you will be able to obtain support when you need it, and it will register you for future updates and announcements.

If you have already used a few Amiga software packages, you can probably skip to the next section, but be sure to make a backup copy of the SAJr. distribution disk first. If you are new to the Amiga, you should know that, while floppy disks are very secure, accidents can happen. It is wise to make a copy and keep the original disks in a safe, dry place, far from coffee spills, magnets, and small animals.

The book Introduction to the Commodore Amiga, that came with your Amiga, tells you how to make a backup copy. When this important step is complete, you are ready to start using SAJr.

If you are an advanced user you may run SAJr. from the Command Line Interface (CLI), but first you must set the CLI's stack to 30,000 or more. When you run a program from the Workbench, the stack is automatically set through the icon. If, when running from CLI, the stack is too small, the program may fail in strange ways.

For users with limited amounts of memory, SAJr. has been made so that it will autoboot. To do this, simply reset the computer, insert the SAJr. disk when the system requests the Workbench, and wait for the SAJr. screen to

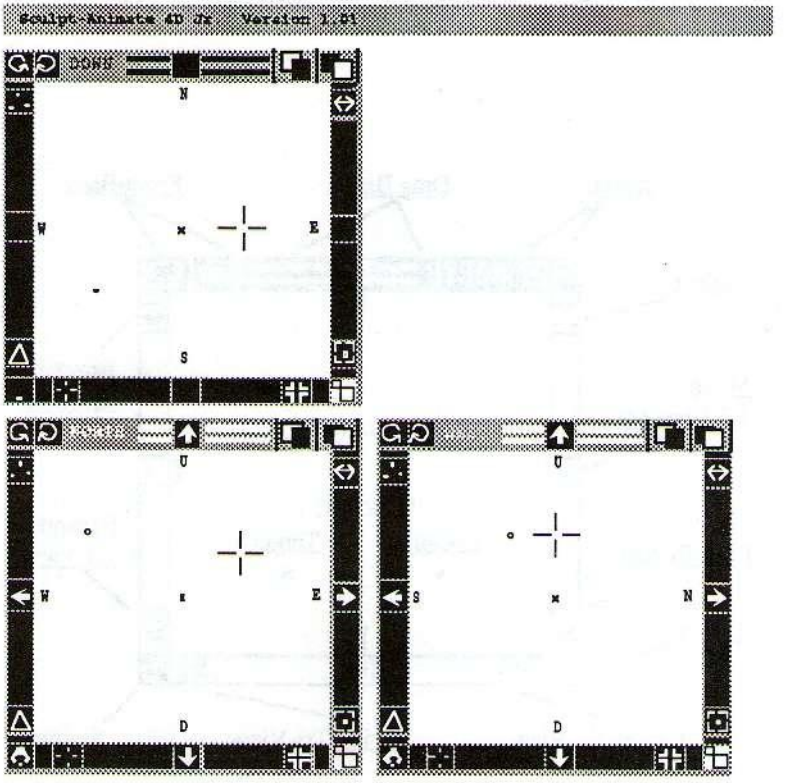
appear. Starting SAJr. this way disables multitasking and the Workbench environment to make more memory available. To save additional memory, use the 'Unload Workbench' and 'Unload Code' menu selections. See the section entitled 'The Menu and Submenu' later in this chapter to learn how to use menus.

The program can also be started from the Workbench Screen by first double-clicking on the icon for the SAJr. disk. Then, when the disk's window opens, double-click on the 'SAJr.' icon that appears.

NOTE: If your system disk (usually 'Workbench') is not in a drive, it is likely that a requester will appear when you run SAJr. asking you to insert the desired volume. This means the program needs to read a file on that disk and can't proceed without it. Whenever this happens, do not click CANCEL. Instead, insert the requested disk and click 'Retry.'

After a short period of disk activity, SAJr. starts by displaying three windows, arranged like three sides of a folded-out box on the screen. These are collectively known as the Tri-View.

The Tri-View

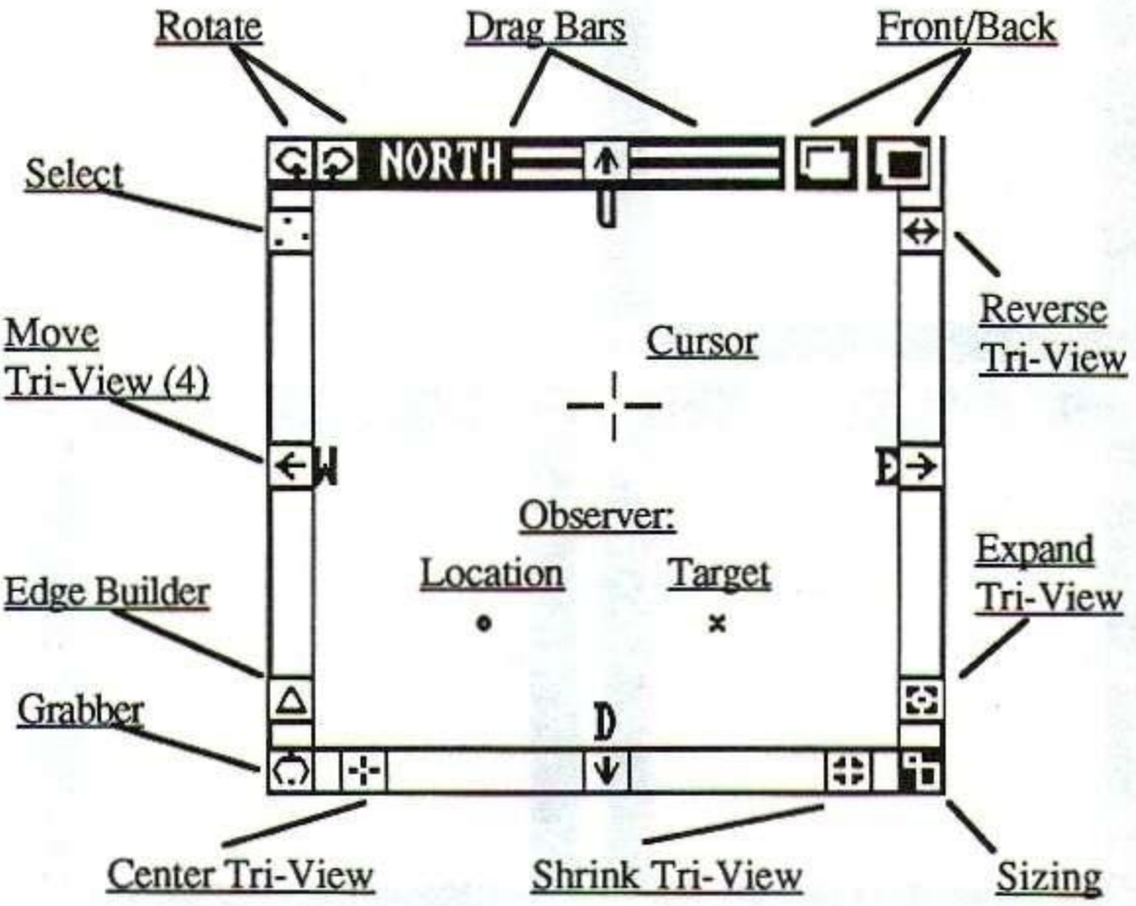


The Three Tri-View Windows

The Tri-View is a necessary part of three dimensional editing, because the Amiga can only display flat images on the screen and SAJr. works with three dimensional solid objects. The three windows show three different views of a rectangular chunk of space. Initially, each window is empty except for a little blue cross in the center, and four letters denoting the cardinal directions that apply to that window.

Look at the window with the name 'North' in its title bar. The North window should be at the screen's bottom left, unless you have re-arranged the windows. This is the

view that faces northward. Notice that there is a small letter just within each border of the window. At the left edge is a 'W' for 'west', to the right, 'E' for 'east.' 'U' for 'up' is at the top, and 'D' for 'down' at the bottom. These are the cardinal directions for this window.



A Tri-View Window

The illustration shows a representative Tri-View window. There are three windows like this on the SAJr. screen. The window at the upper left is a view looking downward, like a map, with north at its top, west to the left, etc. The window at the lower left shows a view looking northward, so west is to the left and up is towards the top of the screen. The third window, at the lower right of the screen, is a view looking west, with south to the left and north to the right of the window. If you have ever encountered technical drawings, or blueprints of a house,

you will be familiar with this way of representing solid objects.

As with most Amiga software, these windows may be moved around on the screen in any way that you wish. To move a window, use the mouse to position the pointer anywhere on the Drag Bar at the top of the window. If you press and hold the left mouse button on this bar, an outline of the window will appear. This outline can be moved anywhere on the screen. When you release the button, the window will be re-drawn at the new position.

The size of a window can be changed by using the Sizing gadget that can be found at the bottom right corner of each window.



The Sizing gadget looks like this.

To change the size of a window, use the mouse to point at the Sizing gadget. Then, hold down the left mouse button while you drag the corner of the window to create a new shape or size. When you release the button, the window is refreshed, but you may notice that something funny happens. Whenever you change the size of one window, one or both of the other windows will change also!

This is because the windows represent a rectangular chunk of empty space. They're like three sides of a box that has been opened out and laid flat. If only one of the sides is changed, then it won't fit when the box is folded back up. One or both of the other sides will have to be changed as well, so the program does this automatically. Suppose that you reduced the size of the box at its south end. The change would also be seen in both the view looking down and the view looking west.

If your windows overlap one another on the screen, you can use the two Depth Arrangement gadgets, the Back gadget or the Front gadget to change which window overlaps the others.



As usual on the Amiga, these gadgets may be found at the right hand end of each window's top border. To use a gadget, point to it with the mouse and click the left mouse button.

The initial sizes of the Tri-View windows are small enough that each window can be displayed without covering another. This is recommended when you first use the program, because it is easy to get confused while working in three dimensions.

When you have gained confidence, you may wish to make the windows as large as possible and use only one window at a time by using the depth arrangement gadgets to go from window to window. The windows can be re-sized manually, but it's quicker to use the command EDIT DO MAKE TRI-VIEW BIG. The complimentary command EDIT DO MAKE TRI-VIEW SMALL will return the windows to their default configuration.

Each Tri-View window also has more than a dozen extra gadgets tucked away in its borders. We will be describing each one in detail later in this book. To avoid confusion, try to restrain yourself from using the gadgets until you know what they are supposed to do. Of course, if you feel adventurous, you can try them anyway. None of them can cause permanent damage to your Amiga.

If you use a gadget incorrectly, a message may appear to warn you of this. Such messages usually display a 'button' gadget, marked 'CONTINUE', beneath their text. To continue with the program, move the mouse pointer into the button's outline and click the left mouse button.

The blue cross shown in each Tri-View window is called the cursor. There is only one cursor, but you see it in all three windows at once. It represents a unique position in three dimensional space. To 'Indicate' a position with the

cursor, place the mouse pointer at the new position in any window and click the left mouse button. The cursor will jump to Indicate the position beneath the mouse pointer. If you hold down the left mouse button and move the mouse, the pointer will vanish and the cursor will move in its place. While the position in three dimensional space that cursor Indicates is unique, the cursor can also Indicate a vertex or edges that lies within a few pixels of the cursor. Throughout this text, we will often ask you to Indicate a position or object in this manner.

You will notice that you cannot move the cursor outside any window. You should also notice that when you move the cursor in one window, it also moves in the other windows. The mouse can only move in two dimensions, but the Tri-View makes all three dimensions readily accessible by switching to another window. We refer to this as the Two Window rule. When a three dimensional position is Indicated in a two dimensional viewing window, the third dimension can only be Indicated in a second different viewing window. You will quickly become accustomed to this style of working in 3D.

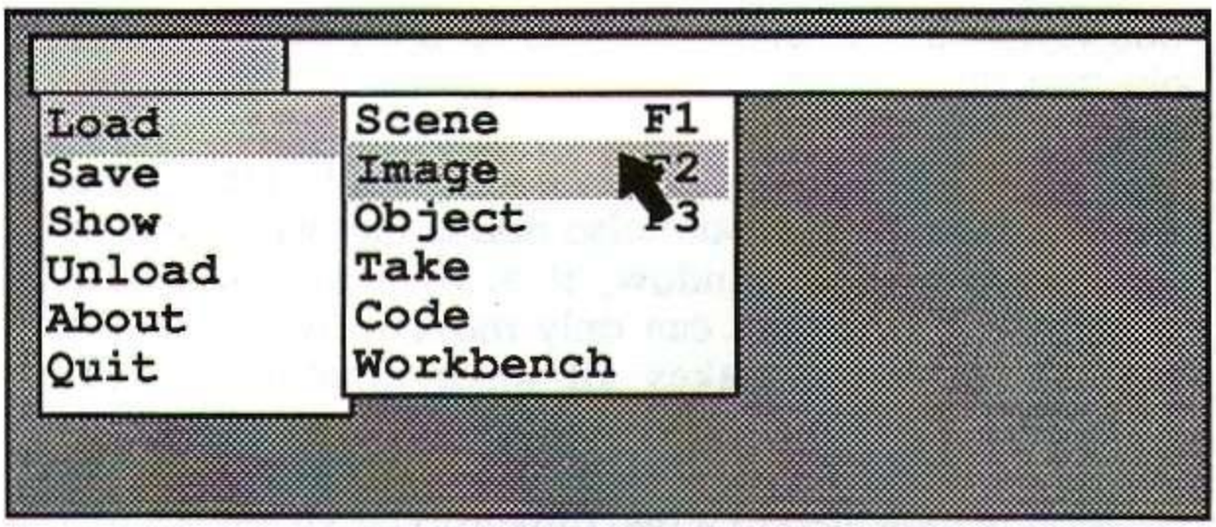
The Menu

Most of the commands that you can issue to SAJr. are made through the Amiga's menu system.

To make the menus visible, press the right mouse button and hold it down. If nothing happens, move the pointer into one of the Tri-View windows, and click the left mouse button (this 'selects' the Tri-View). Then try again with the right mouse button.

The menus appear as a bar at the top of the screen, showing the words 'PROJECT', 'EDIT', 'TOOLS', 'OBSERVER' and 'WORLD'. These are the major menu headings. To see the choices in each menu, move the pointer up to the desired heading, keeping the right mouse button down all the while. When the pointer touches one of the words, a list of choices will spring down.

To activate one of these choices, move the pointer to that item and release the right button. If, as you look through the menu, you decide you'd rather not execute any of the commands, move the pointer away from the menu region and release the right mouse button. Be a little cautious about where and when you release the button, or you may execute a command unintentionally.



Menu and Submenu

As you move the pointer down the list of menu choices, sometimes an additional set of choices will pop up to the right. This is a 'submenu.' To execute a submenu item, keep the right mouse button pressed and move the pointer to the right, then down the new list to the item you want. When you release the button over an item, that action takes place.

At times you may need to issue several commands from the menu, one after another. Rather than releasing the right button and having to reactivate the menus and find your way throughout the menu system each time, there is a way to generate multiple menu commands.

When the pointer is over the menu choice that you want, instead of releasing the right mouse button, you keep the button depressed and click the left button. Still holding down the right button, you can keep moving through the menus, selecting additional commands by clicking the left

button on their entries. When you release the right mouse button, the items you selected will be executed in the order in which they were selected.

There is one exception to this. If one of the items selected brings up a requester (requesters are discussed later in this chapter), subsequent selections in a multiple command will sometimes be ignored. You will quickly become proficient at this speedy way of using the menu system.

For those of you who are not averse to dealing with the Amiga's keyboard, there is an even more fluid and efficient way to get to the menu's commands. In the menu, you will notice odd two-character symbols to the right of some menu items. Each of these is a 'keyboard equivalent' for that menu item. The menu item may selected by holding down the Amiga key to the right of the space bar and pressing the other key shown in the menu.

Another keyboard equivalent is the ESC key. The ESC key will repeat the last menu command.

Loading An Image

To get an idea of the capabilities of SAJr., take a look at some of the images it has created. Hold down the right mouse button and point to the word PROJECT at the top left of the screen. Move down to the word LOAD, and then move to the right to the word SCENE. Finally, move down to the word IMAGE and release the right button. You have issued the command PROJECT LOAD IMAGE.

If you have done this correctly, the menus will vanish and the pointer will be replaced by an image of grinding gears. This indicates that SAJr. is busy and should not be disturbed. In this case, it is looking through the disk to see if it can find any files that contain images. You will notice that the disk drive light is turned on.

After a few seconds, the normal pointer will return, and a new window will be displayed. This window is a 'requester'--the Amiga's standard method of asking the user for more input. There are many kinds of requesters, and this one is asking you to choose the name of an existing file that contains an SAJr. image. This File Requester is used by several commands, so you will be seeing it a lot.

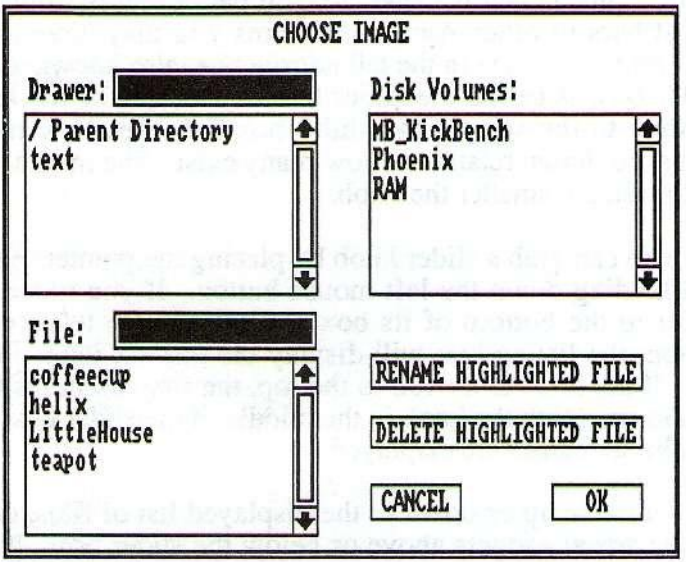
The File Requester contains three large boxes for making selections. The contents of the boxes will depend upon what files you have in your computer.

The box on the upper right is used for selecting a volume. Each floppy disk that the Amiga validates is regarded as a separate volume, with its own name. The RAM disk is also a volume, as are any hard disks, etc. that may be mounted (recognized by the system). A volume is distinct from a disk drive in that a drive may exist that has no volume in it, or you may move a volume from one drive to another. Data resides in a specific volume, but the drive that a volume is mounted in is of little importance to SAJr.

If you wish to inspect the contents of a particular volume, double click on the name of the volume. After some disk activity, the display will change, and some of the contents of that volume will now be visible in the requester's other two large boxes.

The large box at the upper left of the requester contains a list of drawers, if there are any in the selected volume. Volume drawers are used just like filing cabinet drawers. They allow files and information to be organized into manageable categories. You can double click on the name of a drawer to select it. The uppermost entry in this box is called a "/Parent Directory" and it is used to select the drawer, if any, in which the current drawer is stored.

The large box at the lower left contains a list of the names of all the images in the selected drawer on the selected volume. At this time you could double click on the name of the file and the image would be displayed.



The File Requester

Above the Drawer and File boxes are slots where you can type in a name. At the lower right are four boxes named "RENAME HIGHLIGHTED FILE", "DELETE HIGHLIGHTED FILE", "CANCEL" and "OK". These boxes are called buttons--each one can be activated by clicking on it with the left mouse button.

The RENAME button can be used to change the name of a file and the DELETE button to delete it from the system. CAUTION: If you delete a file, there is no reprieve. Its contents will be lost forever.

The CANCEL button is present on most requesters and enables you to cancel the current command. The OK button informs the system to proceed with the current selection.

If there are more than six items to be shown in any of the three large listing boxes, only six items will be shown, but the gadget to the right of each box can be used to reveal more. This gadget is called a scroll bar, and it is similar to scroll bars in other Amiga programs you may have used. The white rectangle in the tall narrow box, also known as the slider box, is called the slider knob. The size of the knob relative to the size of the slider box indicates how many items are shown relative to how many exist. The more items that exist, the smaller the knob.

You can grab a slider knob by placing the pointer over it and holding down the left mouse button. If you move the slider to the bottom of its box and release the left mouse button, the listing box will display the last six items in its list. If the knob is moved to the top, the first six items will be shown. With the knob in the middle of the slider box, the middle six names are displayed.

To move up or down in the displayed list of files, click on the arrow gadgets above or below the slider box. If the listing box contains six items or less, the slider will be inactive, filling its box to show that there are no extra items to display.

Advanced users, especially those with hard disks, may wish to keep different projects in different drawers. This makes it easier to find files and to keep things organized.

When you have found a file that looks interesting, double click the left mouse button on it. The screen will clear and the image will appear one line at a time, starting from the top of the screen. When you wish to stop viewing the picture, click on it first with the left mouse button, and then with the right. If you want to remove the image from memory, hold down the right mouse button, go to the PROJECT menu, move down to UNLOAD, then across to IMAGE and release the button.

Loading A Scene

A scene is a collection of objects and lamps. A scene may also contain information about the observer and a description of the ground and sky. The scene is essentially a description of what composes the image that SAJr. is to make, and how the image is to be made. Scenes are manipulated in the Tri-View and may be stored on disk. Your SAJr. disk comes with some scenes, so let us try loading one of them.

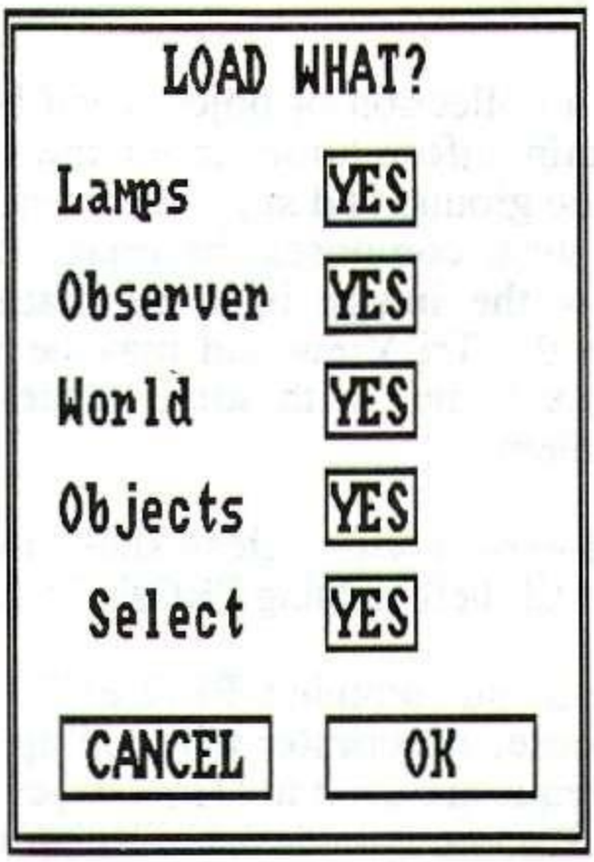
If you want to start with a clean slate, use the command EDIT ERASE ALL before using PROJECT LOAD SCENE.

Execute the menu command PROJECT LOAD SCENE. After a short pause, a requester will pop up that looks very much like the requester used to load images. Scroll the list until you find a scene called "LITTLEHOUSE". Double click on it to indicate that you have made your choice, and that you still want to load a scene.

Objects and lamps that are already in the Tri-View will not be erased when you load in a scene, but at this point a second requester will appear, giving you a choice of what parts of the new scene to load. You may selectively load parts of several scenes to combine them. This is explained more fully in chapter 6.

The 'Load What?' requester contains a list of the scene's various parts. Beside the name of each item is a box containing either the word 'YES' or 'NO'. Initially, each box contains the word 'YES'. If you click on a box with the left mouse button, the contents will change from 'YES' to 'NO' or vice versa. You can click on the boxes until they correctly show what you wish to load. For now, we want to load everything, so you only have to change the boxes if you have fiddled about with them earlier.

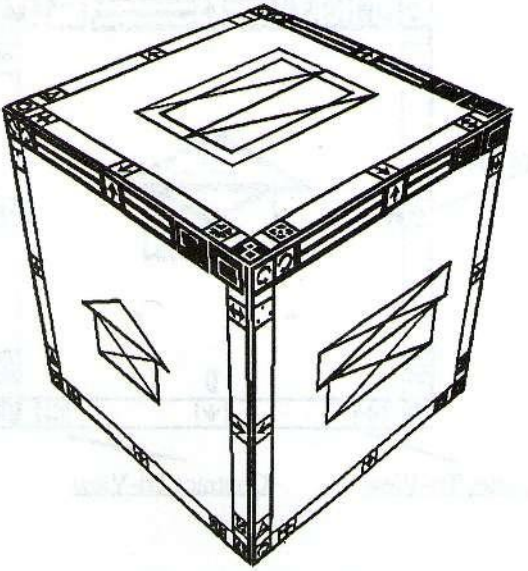
Below the other boxes is one called 'Select' that affects the way that objects are loaded. Normally you will wish to leave it as YES.



The 'Load What?' Requester

When all of the boxes contain the word 'YES', click the OK button. After a few seconds, an image will appear in each window of the Tri-View. At first, the display may seem confusing.

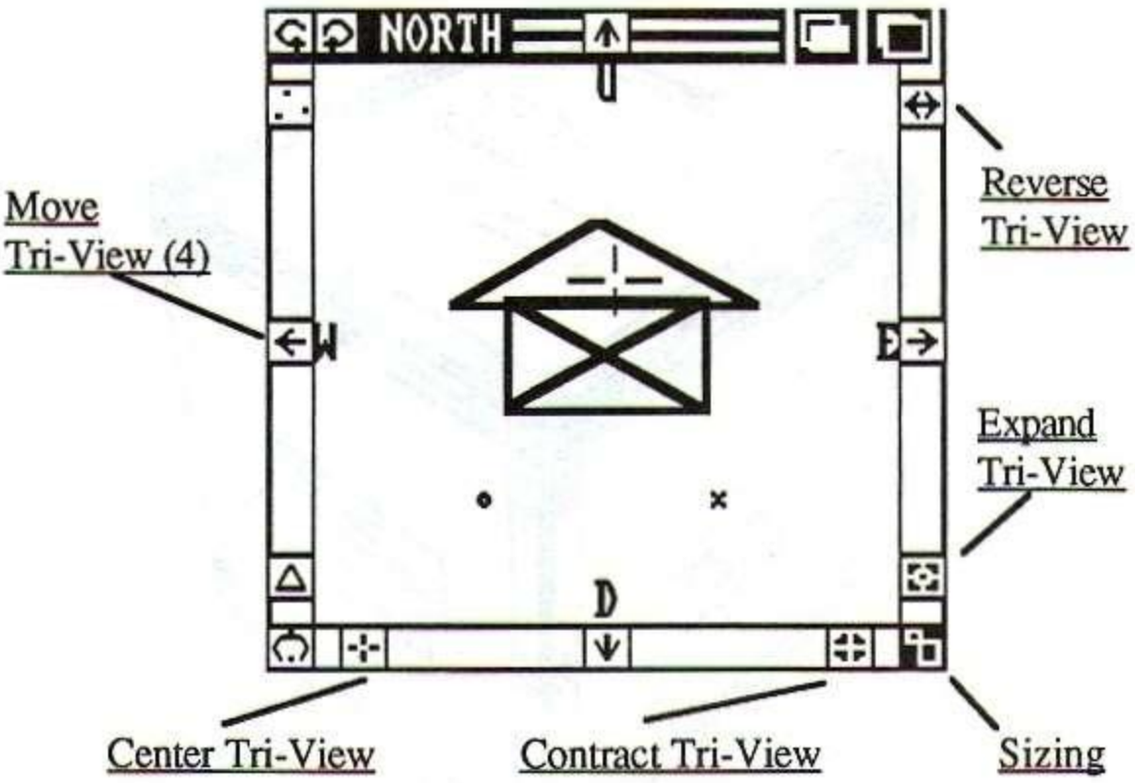
This kind of picture is called a wire frame, because the object is displayed as though it were transparent, with only the edges visible. In addition to the object's outlines, each rectangular side is divided with an extra diagonal line. This is necessary as SAJr. thinks of objects is terms of triangles. Although this seems like a drawback, we will later see how it becomes an advantage when we are building more complicated shapes.



The Tri-View "Box" In 3D.

Let's return to the window with the name 'North' in its title bar. It should be at the screen's bottom left, provided you have not re-arranged the windows. This is the view that faces northward. Remember that there is a small letter just within each border of the window indicating the cardinal directions for this window.

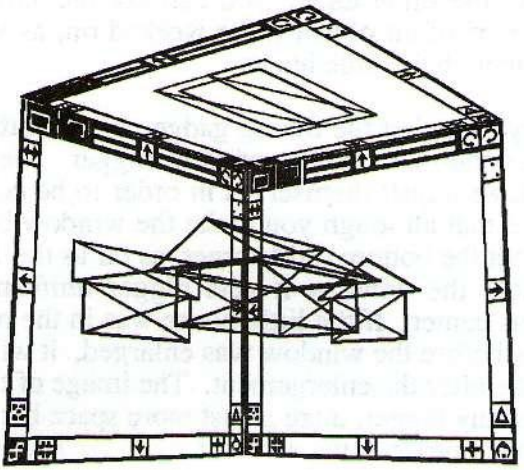
The view in this window should be recognizable as the end view of a simple house with a ridge roof. Note that the wall seems to be braced diagonally in both directions. In fact, the north wall (the one furthest away) is braced in one direction while the south wall (the one nearest to you) is braced the other way. In a wire frame view, you cannot distinguish between near and far objects because far objects are not hidden by those nearby. Now we will look at one way to make sense of these confusing images and at the same time learn about some of the Tri-View gadgets.



The North View

Move the pointer to the 'West' window (the window that looks west, with an 'S' for 'south' at the left side, 'N' for 'north' at the right, etc.). Unless it has been moved, it is the one at the screen's bottom right. Now look at the middle of the left hand border (south, remember?). You will see a gadget with a left pointing arrow. This is called a Move Tri-View gadget. Click on this gadget with the left mouse button. After the usual screen flashing, try to see what has happened.

At first glance, the little house seems to have moved to the right (i.e. north). It's more useful, however, to say that the Tri-View has moved to the left (i.e. south). That is why the gadget's arrow points to the left. It is better to think of the objects in SAJr. as being fixed in space, and the Tri-View as moving. As we said before, the Tri-View windows display three views of a 'box' of space. This box itself can be thought of as a sort of window, into which we can look to see a piece of the Tri-View universe. As you might expect, it makes more sense to move the viewing window around in the universe than it does to move the universe around in the window!



The Tri-View "Box" From Inside

By now you should be able to guess what effects the Move Tri-View gadgets in the centers of the other borders will have. Each will move the Tri-View in the arrow's respective direction.

Back to our example. Look again at the north window. If you are particularly astute, you will have noticed that one of the diagonal braces of the wall has vanished. This is because the brace was part of the north wall, which now lies outside the Tri-View. Click on the gadget in the middle of the North border of the West Tri-View window to move the Tri-View back to its original position. The whole of the house fits inside the Tri-View, so the second brace becomes visible again.

The Tri-View will only display objects, or parts of objects, that are within the chunk of space that the Tri-View represents. When you are dealing with complicated scenes, such as an entire village of little houses, this is a decided advantage. You can move the Tri-View to the area that you are interested in, and not be bothered by neighboring

objects. By the same token, you can use the Tri-View to isolate the part of an object to be worked on, as you have already done with the little house.

Now try dragging the Sizing gadget in the bottom right corner of a window to make a window bigger. See how the other windows adjust themselves in order to be consistent. Also, notice that although you make the window bigger by stretching out the bottom right corner, as far as the Tri-View is concerned, the window it gets bigger uniformly with respect to its center. If the little house was in the middle of the window before the window was enlarged, it will still be in the middle after the enlargement. The image of the house does not get any bigger, there is just more space between the house and the border of the window.

There is another way of changing the size of the Tri-View. Click on the gadget that is near the bottom of the right hand border. The graphic looks like four arrow heads pointing out toward the corners.



The Expand Tri-View Gadget.

Somewhat to your surprise, the Tri-View windows don't expand. In fact, the image of the little house gets smaller. This is relativity once again. What you have done is enlarge the Tri-View's 'box' so that more space is contained within it, even though the Tri-View windows are the same size on the screen. Click on the Expand Tri-View gadget a few times to make the Tri-View even bigger (which makes the house's image on the screen still smaller). Now can you see why we like to think of the Tri-View's volume increasing, even though its windows stay the same size. If the windows had expanded this much, they would have overflowed off the screen. In many ways the Expand and Contract may be thought of as Zoom In and Zoom Out gadgets.

The Expand Tri-View gadget increases the volume of the Tri-View equally in all three directions and its width increases by about 40% with each click. If you hold down the left Amiga key (it is called the Commodore key on newer

Amigas, but we will keep on using the old name) while the gadget is clicked, the amount of the expansion is reduced. Similarly, if the right Amiga key is held down when the gadget is pressed, each click of the gadget will have twice the usual effect. Since the Amiga operating system uses the Amiga's keys for some special functions, the left and right Shift keys perform the exact same task as the Amiga Keys. The left and right ALT keys can be used in the much same way, but decrease or increase the effect of the gadget still more. Several other gadgets, such as the Move Tri-View gadgets, can also be modified by the Amiga keys, shift keys, or the ALT keys.

Let's try moving around in space a little. First, press one of the Move Tri-View gadgets a few times until the house vanishes from view. At first, the cursor (the blue cross) will stay with the house, but when the edge of the window hits the cursor, it will be swept along in the Tri-View.

Next, expand the Tri-View with the Expand Tri-View gadget until the house becomes visible. Point to the middle of the house and press the left mouse button. This moves the cursor to the middle of the house in that window, but look at the other windows to see if the cursor is really in the middle of the house in all views. If not, repeat the operation in another window.

Now, click on the gadget that lies second from the left of the bottom border.



The Center Tri-View Gadget.

This gadget moves the Tri-View so that it is centered on the cursor's location, putting the cursor in the middle of the Tri-View windows. Try holding down the left Amiga key while you click this gadget. The Tri-View is snapped back to its original location and size. This can be useful when you get confused.

Finally, click on the gadget that is second from the right in the bottom border. It looks like four arrow heads pointed to the center of the gadget.



The Contract Tri-View Gadget.

It makes the Tri-View smaller, so the image of the house seems larger.

You have just completed a very useful sequence of commands which you'll use often: Expand the Tri-View to find a point of interest, place the cursor on it, then center the Tri-View and contract it again. Except for short distances, this is often easier than using the Move Tri-View gadgets.

You can use the expand and contract gadgets like the zoom control on a camera. Expanding the Tri-View is like zooming out to a wide angle setting--the field of view gets bigger so objects seem smaller. Contracting the Tri-View is like zooming in for a telephoto effect--the field of view gets smaller and objects seem magnified. You aren't changing the size of things, just changing the way you look at them.

It is easy to get lost in the Tri-View. If you contract it too much there may be nothing visible at all. Similarly, moving the Tri-View too much will move you away from the inhabited part of the universe into a region of empty space. You can always use Expand Tri-View to 'step back' and figure out where you are.

When you run the program, the Tri-View always starts out at the center of the universe, and it always starts out the same size relative to the universe. You can always get back to this setup if you get lost. To return to the center of the universe, hold down the right Amiga key and click on the Center Tri-View gadget. To make the Tri-View its original size, hold down the left Amiga key and click on the Center Tri-View gadget. The Tri-View windows will be re-drawn in their original size and the effects of any expansion or contraction will be removed.

Let us look at one more Tri-View gadget before moving on to more interesting things. This gadget is called the Reverse gadget, and it lies just below the top of the right border.



The Reverse Gadget.

It performs a mirror image swap of the particular window. Alternatively, you may think of it as reversing the direction of view and looking at the opposite side of our 3D box. A north view is turned into a south view, for instance. Our little house is so symmetrical that you may not even notice the reflection, except that the letters indicating the direction near the left and right borders change places. With a more complicated object, this gadget can be more useful.

Making A Picture

If all that SAJr. could do was to manipulate objects in wireframe, it would be no more than a three dimensional computer aided design package. But SAJr. also has image generating capabilities.

SAJr. makes use of a special object called an observer. The observer can be placed at any location in a scene, and can be instructed to look in a particular direction. In addition, the observer can employ one of several imaging techniques. The actions of the observer are akin to photography, and to expand on this analogy, the observer can be equipped with different camera, lens, and film size combinations.

Let us look at how the observer operates by providing instructions to make a picture of the little house. First, we must pick a location from which to view the house. Photographers know that, to take a picture of an object with a regular lens, they must stand far enough back so that the object's image fits within the film size.

To make room for our observer to move back, use the **Expand** Tri-View gadget until the house appears to be about one eighth of the size of the window. Going to the downward looking view (the Down window) Indicate a position (place the cursor with the mouse) that is a reasonable distance from the house. Check in one of the other windows to see that the cursor is at some sensible distance above ground level--about as high as the top of the wall will do fine. Change the Indicated position, up or down as necessary. When the cursor is where you want it, hold down the right mouse button to bring up the menus. Go to the OBSERVER menu and pull the pointer down to the word LOCATION. Release the button.

If you look at the Tri-View closely you will see that a tiny circle has been drawn at the Indicated cursor location. That is the observer! It will be easier to see if you move the cursor away. The observer will stay fixed until you next perform an OBSERVER LOCATION command from the menus, or you load another scene from disk.

The next thing to consider is in what direction the observer's camera is pointing. Although the observer is not equipped with feet to intrude into the picture, you will be disappointed with the picture if the camera is pointing down. Rather than have you specify camera bearings in degrees, SAJr. lets you specify a target position. The observer then points the camera at the target so that the target position lies in the center of the image. A good target would be the center of the house. Move the cursor there, checking in two windows to see that the position is correct. Hold down the right mouse button and select TARGET from the OBSERVER menu. Although you may not be able to see it for all the clutter of the house, a small x will be drawn at the target location.

For the observer's camera to record an image, there has to be light. Some background illumination is provided as a default, but this, by itself, gives a flat, uninteresting image. The primary light source in most SAJr. scenes is one or more lamps. To put a lamp in a scene, place the cursor at the location where you wish the lamp to be--somewhere near the

observer and a little to one side and above will usually work well. Then, hold down the right mouse button and select LAMP from the EDIT ADD LAMP submenu. A small white star should appear at the cursor's position. Lamps shine their light in all directions, so you don't have to worry about pointing them, and the camera's exposure setting will automatically be adjusted to account for the extra light.

Now that you have placed your observer, target, and lamps, you can instruct the observer to start taking a picture. Go to the OBSERVER menu and select START. A small picture will appear near the bottom of the screen and after a few seconds will start 'developing'. If you have not accidentally changed any of the observer's attributes, then the picture size will be MEDIUM and the image type will be SCANLINE. These settings will be explained a little more in this chapter, and fully in the next.

While the image is rendering ('developing'), and after it's finished, you may want to move it out of the way. You can do this as follows. Point to the picture and first press the left mouse button. Release it and press the right mouse button. The picture should vanish.

The picture is still in the computer's memory, however, and can be recalled by going to the PROJECT menu and selecting SHOW IMAGE. This brings the Amiga 'screen' on which the image was rendered 'to the front.'

Let us try a different image of the same scene. Go back to the OBSERVER menu and select MODE WIRE FRAME. This mode renders much more quickly than SCANLINE, so we can afford to make the image larger. Go to the OBSERVER menu again and select IMAGE SIZE FULL. Finally, select OBSERVER START.

The image that you see will show all of the edges, but it will be drawn in the correct perspective, from the Observer's standpoint.

Saving An Image

The image that you have just rendered can easily be saved to disk. Hold down the right mouse button and go to the PROJECT menu, follow down to SAVE IMAGE then across to IMAGE and release the button. A File Requester will appear on the screen, asking for the name that is to be used when the image is saved on the disk.

If you want the data to go to a particular volume or drawer, click on the appropriate names in those listing boxes. You will probably want to use a new name for this file. Next to the 'File:' heading over the file listing box is a gadget called a 'text entry gadget.' You may click on this gadget and then type in a new name for the image. When you hit the RETURN key, the image will be saved.

If you want to save over an old image, just double-click that image's name in the file listing box. Since the file already exists, the program will ask you to confirm that this is really what you want to do. Saving data using an old name will cause the old data to be destroyed.

Images are saved in a standard 'IFF-ILBM' file, which may be loaded into some other Amiga programs, such as paint programs, etc.

Images are not given icons. You may copy, list, and delete images using the Amiga's CLI interface. All image files are automatically given the name extension '.image' by SAJr. This is how the program keeps track of which files are images, which files are scenes ('.scene'), etc.

Saving A Scene

The contents of the Tri-View can be saved to disk with the command PROJECT SAVE SCENE. This requester works the same as the SAVE IMAGE requester. After choosing a file name with this requester, a second requester will appear, asking you what parts of the scene you wish to save. This is very much like the requester that appeared

when you loaded the little house. Usually, you will just leave all the upper gadgets set to 'YES' and the 'which?' gadget set to 'ALL.' The meanings of these gadgets will be described in Chapter 6.

Leaving SAJr.

If you wish to perform other tasks while SAJr. is still running, you can push its screen back with the Back gadget at the top right of the screen. You are then free to use the Amiga's multitasking capability, as long as your machine has enough memory and has a Workbench running or an available CLI.

Be sure not to forget that SAJr. is running, however! You will waste memory if you start it twice. You may also run out of memory if you run SAJr. with one or more other large programs.

To exit SAJr. completely, execute the menu command PROJECT QUIT. Because you may have some valuable data, scenes or images, that you have neglected to save to the disk, SAJr. will ask for confirmation before actually stopping execution.

If You Have Problems

As with any program, you may occasionally run into trouble. This section attempts to provide some guidance on how to deal with difficulties, should they arise.

If you have been successfully using SAJr. up to this point in the text, then you have already cleared some major hurdles--especially if you are new to computers. Your Amiga is plugged in and connected together properly and probably free of any grossly debilitating defects. You have mastered the boot-up process, the Workbench screen, and the vital process of backing up floppy disks. Furthermore, you have managed to run SAJr. and operate some of its menus and other controls, and have grown somewhat

accustomed to the style of this manual. In short, you have gained a valuable foothold on the rocky slope of the learning curve, and you can skip right now to the next chapter.

If, on the other hand, your efforts so far have not been entirely successful, do not be too discouraged. Rest assured that many people have been using this program and its predecessors for some time and with good results. Any problem you may be having is only a setback, not a defeat. Most people involved with three dimensional modeling spend weeks studying manuals and attending courses in order to gain the proficiency that you can acquire in a few hours with SAJr. If you do have problems it is likely to be due to the intrinsic difficulty of the subject, not to any shortcoming on your part. Be assured that if you persist you will soon have mastered a valuable new set of skills. What follows is essentially a short course on troubleshooting any computer problem, as applied to SAJr.

If your Amiga is simply not working, don't fail to check the obvious. Be sure all of its connectors are properly plugged in. Even the most experienced computerists have, at one time or another, been tripped up by a loose cable. Other problems could be caused by a bad system disk, or perhaps even a hardware problem. Don't hesitate to call your dealer if you can't find the answer in your Amiga's manual.

If you have trouble backing up the SAJr. disks, or if the program won't load, it could be a problem with either the distribution disk or with your Amiga's disk drive. The first thing to try is using the disks in another drive, either your second drive or a friend's drive. If the program works in another drive then the hardware is probably at fault. Otherwise, the program disk may be flawed, perhaps because of heat, cold, X-ray or magnetic damage incurred in shipping. If the disk seems to be at fault, read the warranty statement that was enclosed with the program. It will explain how to obtain a replacement.

Finally, if the program runs, but doesn't behave as you expect, there are a number of things that could be wrong. We will consider these one by one.

Of course, there is the possibility of bugs (errors) in SAJr.'s documentation or software. Few, if any, programs escape this problem entirely, but it is always the goal. Every effort has been made to ensure that SAJr. is bug-free, and we continually hunt for bugs.

If you encounter a situation in which the program behaves in a confusing manner, contrary to the documentation, please write down step-by-step instructions telling how to re-create the problem and send them to Byte By Byte (Attention: SAJr. Bug Reports). Be sure to include specifics about your Amiga's model, peripherals, enhancements, and system software. Also note the SAJr. version number you are using. This is displayed at the top of the screen when you run the program.

We would also like to hear of any changes or additions you would like to see made to the program (Attention: SAJr. Enhancements). Updates and new releases will be made available periodically.

Before you file a bug report, read the remainder of this section. It describes the majority of problems that users have had in the past and could provide a quick solution to your particular problem.

First, the obvious. It is very possible that the program simply doesn't do what you expected it to. SAJr. is a remarkable program in many respects, and it does things that have never been done on personal computers before. To do these things, it must behave differently than, for instance, a paint program. The nature of solid object modeling on microcomputers makes it impossible to actually 'draw' in three dimensions, the way you would draw on paper.

If you encounter confusing behavior, read the related part of this manual carefully, remembering that every tool was designed for a specific purpose, and with the Amiga's

limitations in mind. Even if result of a function is not quite what you expected, it is generally the best or only solution to a difficult problem. Never forget the value of simply doing something else for a while--the problem will usually seem simpler the second time you look at it. If you simply cannot fathom some aspect of the program's use, consult the support policy statement that is enclosed with the program for information on how to obtain special help.

At times SAJr. may seem to take an inordinate amount of time to do something. This is because different scenes in SAJr. can present entirely different problems to be solved and it is difficult to assign a 'proper' amount of time to complete the action. Extremely simple scenes, such as the example in this chapter, should never take more than a few minutes for any operation. If you increase the size and complexity of your scenes gradually, then you will surely develop an idea of how long things should take. The following chapters explain some of the factors that can make a scene develop more or less quickly.

There are also times when the amount of RAM or type of memory in the system is inadequate for a particular task. This situation is described more fully in the next chapter ('Memory usage'). Also, users working with floppy disks will soon wish for more spacious storage media. Chapter 9 has some information on how to do without.

A more insidious problem can occur if the program 'crashes' or 'hangs up'. Again, this could be a hardware problem, especially if it seems to happen at random, and it is good to try the program on one or two other Amigas if possible. SAJr. exercises the Amiga hardware as much as any program possibly can, using more memory and performing more intensive calculations than had previously been the norm (it has even been used to 'acid test' memory board designs). If any part of a computer or peripheral is 'marginal', then it is more likely to fail when running SAJr. than when running most ordinary software.

If the crash or hang-up always occurs when you perform a particular task, something might be wrong with the SAJr.

program file itself, even if the program starts up OK. It is certainly worthwhile to try making another working copy of your original disk, although this will not help if the original itself has been corrupted. Again, see the warranty information for replacement instructions.

Crashes are certain to occur if you run SAJr. from the CLI without first setting that CLI's stack to about 30,000 bytes (or more), or if you use the wrong version (before 1.2) of the Amiga's Workbench and Kickstart disks. Problems can also occur if system files on the Workbench disk are somehow corrupted, so its a good idea to try booting with a different copy or reverting to a copy of your original system disks. Some resident 'enhancement' utilities, especially the public domain ones, can cause problems. Try and make your environment as standard as possible. In particular, avoid running 'screen and window control' or 'mouse enhancement' utilities when running SAJr.

It might seem alarming that there are so many things that could go wrong when you are using SAJr., but these are only things that could go wrong, and usually do not. The program and its predecessors have been intensively tested for literally thousands of hours at Byte By Byte, and have been used by thousands of people. The single most often reported 'problem' with SAJr. is that users can't keep their hands off of it, it is so absorbing!

Notes

CHAPTER 3

THE OBSERVER

In the last chapter you were briefly introduced to the observer, a feature of SAJr. that controls how images are produced. In this chapter we will look at the observer in more detail and describe the different kinds of images that can be produced.

Before you start to construct complicated objects, it is useful to understand how to make images. You will want to make make images frequently during a construction phase, in order to be sure that you have built exactly what you intended. Although the Tri-View provides everything you need to construct an object, it is almost always best to generate an image to check your progress.

Imaging Modes

In order to record images, a real-life observer has to have a camera, and the camera needs film. What kind of film determines the quality and resolution of the resulting image, as well as how quickly it can be captured. Although your computer doesn't use film, there are still a number of different ways for it to 'take a picture' of the objects in your scene. These methods all differ in speed and realism, just as various types of film do. The MODES submenu in SAJr. lets you set the kind of 'film' the SAJr. observer will use.

The first mode in the menus is WIRE FRAME. This is the highest-speed rendering mode, but the lowest in quality. Wire Frame yields two-color images which closely resemble the objects seen in the Tri-View, with one significant difference. While Tri-View windows display a flat 'orthographic' projection of their views, Wire Frame shows you a true 'perspective' view, as seen from the observer's viewpoint. Only edges are shown, but each edge is shown

in the same place it will appear in the other modes. This makes Wire Frame an excellent mode for taking a look through the observer's 'viewfinder' before rendering in a slower mode. The command EDIT MODIFY WIRE-FRAME COLORS can be used to change the colors used in wire frame renditions.

The other two modes used by SAJr. render objects as solid shapes in multiple colors. The fastest way to do this is to display each visible portion of the objects, with color and shading determined by the specified light sources. SAJr. calls such an image a SKETCH. This mode includes the effects of perspective and the resulting pictures look reasonably solid and realistic.

As we shall see in Chapter 4, the objects in the world of SAJr. are made up from triangular pieces. In the SKETCH mode, the image of each piece is 'painted' onto the screen in one operation The order in which the faces are painted is simply determined by how far they are from the observer. There are circumstances where a simple distance does not suffice to provide a correct order for the faces. Sometimes one object that should be hidden behind another will be incorrectly drawn in front.

The second solid color modeling mode is the SCANLINE mode. This mode yields results that look similar to SKETCH, but it generates the image one line at a time. Although this can take longer than generating the image in SKETCH mode, each face will be correctly rendered, even if one face intersects another.

Depending upon which mode you choose to use, the time that it takes to generate an image depends upon the complexity of your scene. The time taken in the SCANLINE mode also depends upon the size of an image. So, if you have a complicated scene, it can be more efficient to make a small image before making a full sized one in order to check that everything is the way that you want it.

Now that you know about the various imaging modes, you need to know how to select one. You can specify which

is to be used with the OBSERVER MODE menu command (from now on in this book, we will assume that you have mastered the operation of the menu). The MODE submenu lists the choices with a checkmark to show which mode is currently active. If the marked item is not the one you want, select the correct menu item in the usual way. The checkmark will move to reflect the change.

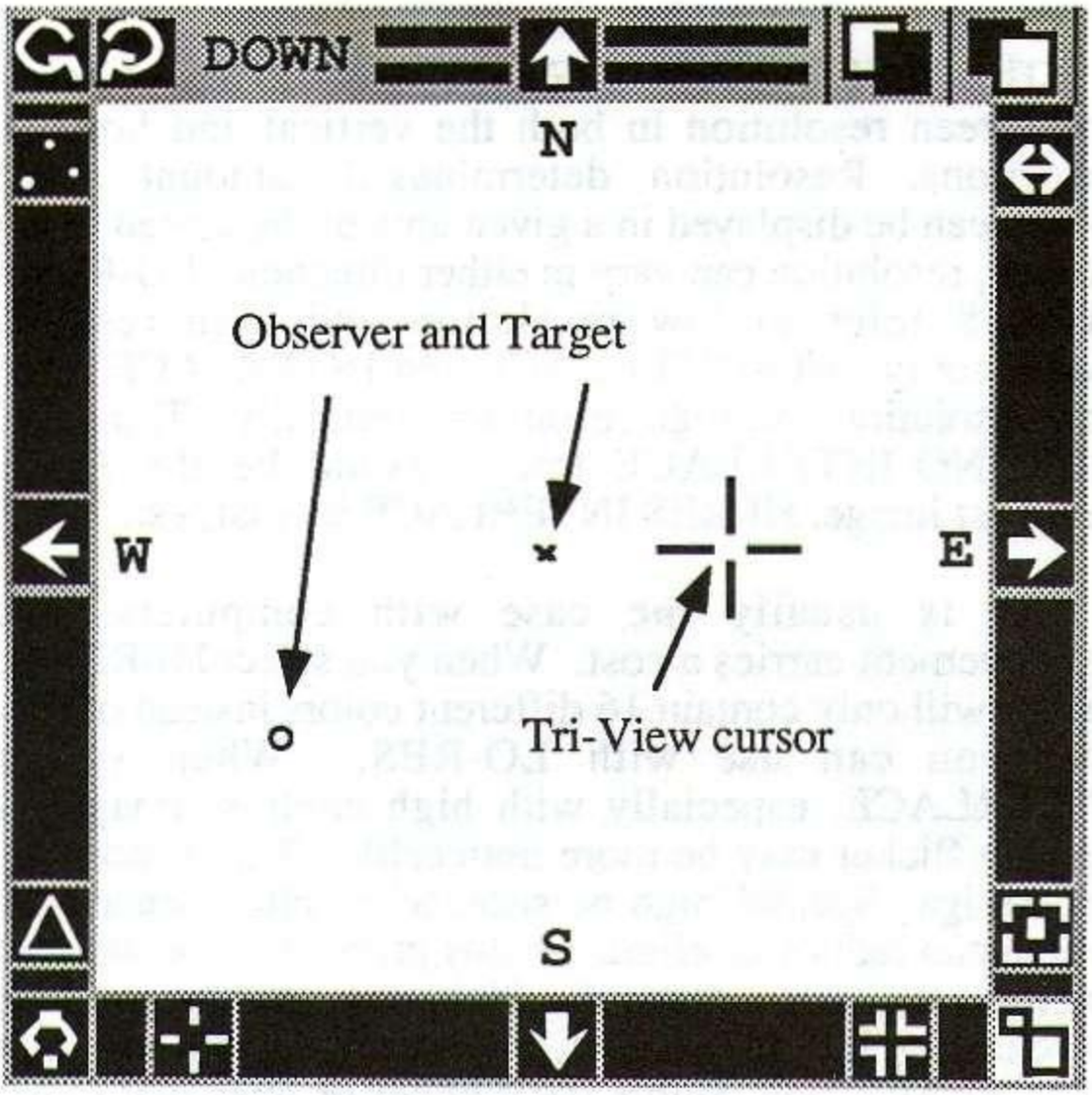
The OBSERVER MODE command also lets you choose the screen resolution in both the vertical and horizontal directions. Resolution determines the amount of detail which can be displayed in a given area of the screen. On the Amiga, resolution can vary in either direction. LO-RES and HI-RES refer to low resolution and high resolution horizontally. NO-INTERLACE and INTERLACE refer to low resolution and high resolution vertically. Thus, a LO-RES, NO-INTERLACE image would be the Amiga's grainiest image, HI-RES INTERLACE its sharpest.

As is usually the case with computers, every improvement carries a cost. When you select HI-RES your image will only contain 16 different colors instead of the 32 that you can use with LO-RES. When you use INTERLACE, especially with high contrast images, the display flicker may be more noticeable. This is normal on the Amiga. Special 'high-persistence' monitors are available which can reduce or eliminate this problem. Special screen overlays can also reduce the flickering, but try one before purchasing it. In SCANLINE mode, when you select INTERLACE you double the number of lines that must be generated and an image will take almost twice as long to generate.

Placing The Observer

In the last chapter, we explained how to change the observer's position, but we will repeat it again (in case you look here first). First you must Indicate a position in the Tri-View with the cursor. You may need to make the Tri-View bigger, so you can place the observer far enough back to get a reasonable view.

When the cursor is at the right place (look in at least two windows, to be sure), go into the menu and select OBSERVER LOCATION. When you execute this command, a small circle will appear in each window, to mark the observer's new location.



Observer and Target

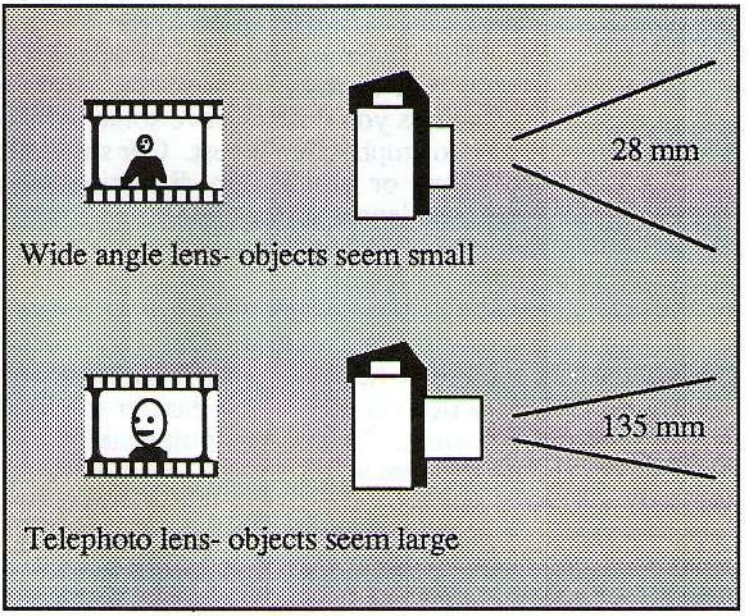
Aiming The Observer

The observer must be told which way to look. Although you may know what is the most interesting object in the world of SAJr., the observer doesn't. As we explained in Chapter 2, the observer always points towards a special spot called the target. This is specified in a similar manner to the

way that we picked the observer's location. We move the cursor to show the target position and select OBSERVER TARGET from the menu. A small x will mark the spot. The target is usually placed near the most interesting part of the scene, so that part will appear in the center of the image.

Camera Lenses

We have already drawn an analogy between the observer and a photographer wielding a camera. If the camera is equipped with a zoom lens, it can zoom in for a telephoto shot where a small angle of view fills the whole picture. Alternatively the camera can zoom back so that the field of view is wide. As with a camera, the observer can be supplied with a lens of your choosing. Use the OBSERVER LENS menu command to examine and change the current selection.



The Effects Of Different Lenses

The observer has three standard lenses that should suffice for most scenes. Alternatively, you may select the OBSERVER LENS SPECIAL option. When you select this, a requester will prompt you to enter the focal length of a lens of your own design. The smaller the number that you enter, the wider the angle of view, and the smaller an object will appear. For instance, of the three standard lenses provided, the NORMAL setting corresponds to a SPECIAL setting of 50, WIDE-ANGLE corresponds to 28, and TELEPHOTO corresponds to 135.

In general, these numbers are analogous to millimeter focal lengths of lenses on a typical 35-mm camera. If you enter a number in the telephoto range of 135 or larger, distant objects will appear larger than normal, with a reduced sense of perspective. If you enter a very small number, such as 10, then you will get a fish-eye effect with grossly exaggerated perspective.

Astute photography buffs will notice some differences between the way that images are rendered and the way that real cameras work. The observer does not have a focus control, so everything will always be in focus. This is usually an advantage, unless you wish to have some parts of a scene blurred in order to emphasize the rest. Our simulated lens also lacks the barrel or pincushion distortion from which all but the finest real lenses suffer.

Exposure Control

Most modern cameras have an automatic exposure control so that pictures don't come out too dark or too light. SAJr. simulates this feature. This means you usually don't have to worry about exposure when you add or move lamps in a scene.

Like the automatic exposure control on a camera, most of the time SAJr. does a good job of controlling the brightness of the image. It works according to a simple rule. The object in a scene that is closest to a lamp will be brightly lit, but not lit so brightly that it washes out.

Sometimes, perhaps for artistic reasons, you may want to change the exposure. Select OBSERVER EXPOSURE MANUAL and a requester will pop up asking you to override the automatic setting. Enter a number as a percentage of the standard value. If you want your picture to be twice as bright as normal, enter 200. Because the Amiga screen has a very limited range of brightness, the image that you get may look flat and washed out. Similarly you can enter a number less than 100 if you desire a gloomy and somber effect.

Image Size On The Screen

You can select the size of image that SAJr. creates. Usually you will want the image to occupy the full size of the screen, but there are a few reasons why you may want to make a smaller image.

If you are planning to import your picture into one of the Amiga paint programs, then a small picture may be all you need, particularly if you are using the image as a brush.

Another advantage of a small image is that you can continue to work with the Tri-View, with the picture displayed at the bottom of the screen. If you point to the screen near the top of the black area to either side of the picture, and hold down the left mouse button, you can drag the image up the screen or push it down so that it almost vanishes. To do this, you have to hit the black area just right, between about four and twelve scan lines from the top.

The best reason for generating small images in the SCANLINE mode is to reduce the time that it takes to create them. It is almost always a good idea to try a small image to ensure that everything is correct before starting a full size picture.

Take note, however, that a certain part of the rendering time for any scene will be the same, no matter what the image size is. Specifically, the 'grinding gears' period--before SAJr. actually begins to draw the image--will not be

reduced. At that time the program is processing the scene data, which will not vary with the image size.

You change the picture size by selecting a choice from the OBSERVER IMAGE SIZE menu selections. Note that there are versions of SAJr. available for two different video standards: one for NTSC, which is used in the USA and Japan; and SAJr.(PAL) for PAL, which is used in Europe and elsewhere.

Here are the image sizes (in pixels) that may be used:

Size	SAJr.(NTSC) width x height	SAJr.(PAL) width x height
MEDIUM	160 x 100	160 x 128
FULL	320 x 200	320 x 256
OVERSCAN	352 x 220	352 x 282

The above values are for low resolution non-interlaced images. If INTERLACE is employed, there will be twice as many vertical pixels, but the image size will be the same. If HI-RES is used there will be twice as many pixels horizontally.

Tilting The Camera

Many photographers have difficulty holding a camera level. You have doubtless seen photos of perfectly upright buildings that resemble the Leaning Tower of Pisa, or of people who seem to have lost one shoe. In SAJr., the observer always holds the camera straight for you.

Most of the time it is easy enough for the observer to know how to orient the camera. But what if the observer is directly above the target? No one camera orientation makes more sense than any other in this situation, because 'up' isn't in the picture! In this case, the program will pick an arbitrary orientation.

What is more likely to happen in this situation is that the camera will be almost, but not quite, directly over the target.

If it is a little bit east of the target, then 'up' will be west. If it is a little bit northwest, 'up' will be southeast. In other words, 'up' is always the direction opposite the observer's position. In an animation, this can make the image seem to wheel about drunkenly. A similar behavior occurs when the observer is beneath the target.

If you must place the observer above the target, it's fairly simple, but you will need to do so carefully. As we've seen, if the observer is to look downward, the orientation of the resulting image can be controlled by placing the observer just a little bit to one side. Only a tiny offset will be needed, so the image will be indistinguishable from a true downward view.

If you're surprised that 'up' and 'down' can cause us so much trouble, don't be. We're in good company, some of the largest telescopes in the world have trouble looking at the celestial pole, for similar reasons!

Starting The Image Making Process

After you have set up a scene and adjusted the observer's attributes, you are ready to start the picture making process with the OBSERVER START command from the menu. If a hidden image is already being rendered (see the next section), you will be informed of the fact and the scene will not be rendered. Otherwise, a blank screen will appear and the image creation will commence.

Image creation will proceed differently, depending on the rendering mode and whether the image is being rendered via OBSERVER START or as part of an animation. SAJr. will sometimes choose not display the image as it is being rendered. This conserves memory and can reduce rendering time.

Hiding And Restoring The Image

Once a picture has been created, or sometimes even as it is being produced, you can hide the image from view. This will only be possible if the normal system pointer is visible, not if the grinding gears are going.

To hide the image, first place the pointer over the image itself and click the left mouse button. This will make the mouse pointer disappear, so don't move the mouse any more or it might go off the picture. Now if you click the right mouse button the image will disappear. You will usually just want to click left-right in quick succession.

When the image disappears, the Tri-View will become visible. Before you proceed, you must move the pointer inside one of the Tri-View windows and click the left mouse button. This 'selects' the Tri-View, which makes the menus and keyboard active. The best place to select the Tri-View window is in the drag bar. If you click somewhere else and a tool is active, you might accidentally alter your objects!

You can bring the hidden image back into view with the menu command PROJECT SHOW IMAGE. If you want to remove the image entirely, freeing up the memory it uses, use the menu command PROJECT UNLOAD IMAGE.

Experienced Amiga users will find that the hidden image is on an Amiga graphics 'screen', behind the screen that contains the Tri-View. There can be several such screens present in the system at any one time, and there are a number of ways to bring a particular screen into view (see your Amiga documentation for details). SAJr.'s OVERSCAN sized images are larger than the Amiga's standard screen, however, so they will not be properly displayed unless you use PROJECT SHOW IMAGE to bring them into view.

Aborting The Image Making Process

When the observer is instructed to start making a picture, there is a period when the full capacity of the Amiga is

required. During this time, the pointer will turn into the grinding gears image. As soon as the regular pointer returns, you are free to continue working with the Tri-View, even though image generation is still in progress. Changes that you make to the scene or its objects will not affect the image being generated.

If, after your normal pointer returns, you decide that you do not want the image, you can use the OBSERVER ABORT menu command to stop the processing.

Memory Usage

If you are curious about what resources are being used, you can execute the command OBSERVER STATUS. This turns on (or off) a display at the top of the screen. Most of the items in this display are explained in the next section. On the left side of this display, the amounts of unused chip and fast memory are shown. Chip memory is required to display screens, while both chip and fast memory can be used for storing code or object data. This distinction will be explained more in a moment.

When memory gets low, the program may not be able to proceed. Sometimes SAJr. will recognize the condition and simply refuse to perform a command. At other times the memory shortage occurs at a critical point and there is no alternative but to halt the execution of the program without warning.

It is a good idea to keep track of memory usage, and save any important scenes or images at frequent intervals. SAJr. is a large program, but all the code need not be in memory at once. When memory starts to run low, unused code is jettisoned to allow space for more important items. When memory becomes available, the code will be reloaded in anticipation of being used. You can control this automatic operation with the PROJECT LOAD CODE and PROJECT UNLOAD CODE commands. The only drawback of having code unloaded is that a delay will occur before some commands can be executed, as the code must be recovered

from disk. If you are working with floppy disks, you will be asked to insert the program disk when this happens.

This is not the only time when SAJr. will load code. The program makes use of a facility of the Amiga called 'disk loaded libraries'. Most of the time you won't need to worry about this, but if the you are ever asked to insert the Workbench disk (or whatever disk you used to 'boot up' the system), this is probably the reason. Of course, other programs running concurrently, including the workbench screen, may also ask for a particular disk. This is normal on the Amiga.

Just as the code can be loaded and unloaded, so can the workbench screen. The commands PROJECT UNLOAD WORKBENCH and PROJECT LOAD WORKBENCH do this. Note that the workbench can only be unloaded if it is not being used, and that it can never be unloaded if a CLI window is present. PROJECT UNLOAD WORKBENCH will do nothing in these cases.

The terms 'chip memory' and 'fast memory', used above, refer to Random Access Memory (RAM), this is volatile (non-permanent) memory within the computer which can be accessed directly by the microprocessor.

Fast memory is the Amiga's general purpose memory. All expansion memory on current Amiga models is classified as fast because, theoretically, the microprocessor has free access to the memory at all times. Fast memory is where programs and object data are stored whenever possible.

Chip memory may also hold such information, if there is no more fast memory available. More importantly, though, chip memory is the only place where graphics display data can be used, since it is the only memory which the Amiga's graphics chips may 'borrow' from the microprocessor. As you can imagine, this makes chip memory very precious on a graphics-intensive machine like the Amiga.

Current models of the Amiga can have no more than 512K of chip RAM. This means that, no matter how much

expansion RAM you have, there will still be times when an operation cannot be performed. When you generate images with HI-RES, INTERLACE, or OVERSCAN set, you may need to make it a habit to free as much chip RAM as possible. To free the memory, do an UNLOAD WORKBENCH and make the Tri-View windows as small as possible.

Other Status Display Items

As the last section explained, the command OBSERVER STATUS calls up a program status display. In addition to the 'free' memory readout already discussed, the status display also lists how many vertices (selected and unselected), edges, faces and lamps are in use. If an image is being rendered, a rough estimate of the remaining time is also displayed.

The vertex count readout can be extremely useful. It is easy to count vertices just by selecting an object, or part of an object, and looking at the status display. The edge count and face count readouts are primarily useful as an indicator of how complex the scene is. The more faces there are, the longer the scene will take to render. It is a good idea to be careful not to use any more faces than necessary.

Notes

CHAPTER 4

BUILDING OBJECTS

In this chapter we will describe the way that SAJr. represents solid objects, and how you can design any shape with your mouse.

Objects in SAJr. are built up from individual triangles. In the Tri-View, each triangle is visible as three edges joining three vertices. In this chapter we will describe the most fundamental level of object building: dealing with the triangles on an individual basis.

It would be extremely tedious to build a complicated object by constructing all of its triangles one-by-one. In Chapter 5 we will be using some power tools that enable you to conjure up complicated shapes with hundreds of triangles, using only a few strokes and clicks of the mouse.

You need to know how the low-level construction is performed, though, to be in a better position to use these power tools. The low-level operations can also be used to expand upon objects that are hewn by more powerful techniques. This combination of methods greatly increases the total number of objects which may be easily created.

Vertices

The simplest entity that SAJr. can handle is called a vertex (the plural of vertex is vertices). Vertex is just a fancy name for a point at the corner of a triangle. We had better not call the vertex a point, though, because that word is already over used. Otherwise, we might find ourselves forced say 'point the pointer at the point!'

If you remember how we located the observer within the Tri-View, you will not be surprised at how we create a

vertex. First we position the cursor at the exact spot where we would like a vertex (by pointing with the mouse and clicking the left mouse button). Then we issue the menu command EDIT ADD VERTEX.

There is an easier way to do this. Just move the cursor to the correct place and, while still holding down the left mouse button, click the right mouse button once. Instead of bringing up the menu, clicking the right button with the left button down will create a new vertex. It is portrayed in the Tri-View by a bright yellow dot.

Try making some vertices, but at this stage do not make more than three of them, we will see why in a moment. Create the three vertices in a triangular pattern about half the size of the Tri-View.

Edges

The next step in low level construction is to build edges. Edges always end at a vertex. Indeed, the main purpose for having vertices is to define the position of edges.

If you succeeded in creating three vertices, there is a Tri-View gadget that you can use to join the vertices. It is located near the bottom of the left hand border of each window.

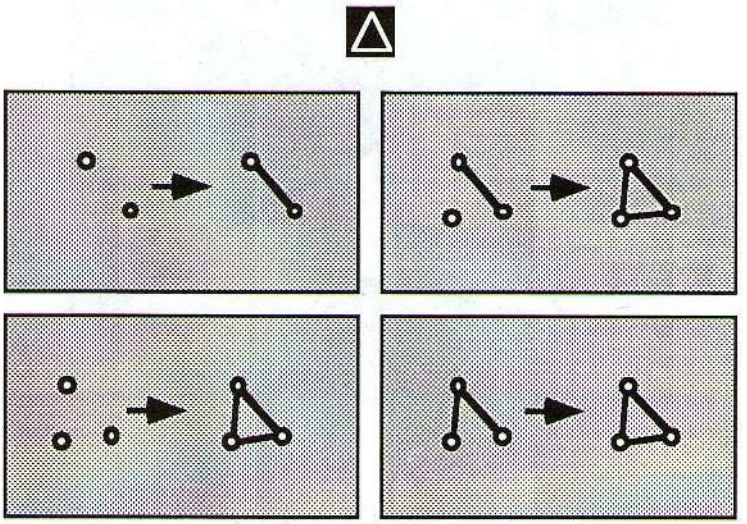
Its icon is a triangle like this .

Click on it and see that dark lines are drawn between the vertices. This is how the Tri-View portrays edges.

If, for some reason, edges were not created, and the screen simply flashed at you when you clicked on the Edge Builder gadget, it was probably because you had created more than three vertices. This can happen if you accidentally double-click the right button, creating two vertices in the same spot instead of one, or if you have already been playing with the mouse. You can clear away the vertices by

using the EDIT ERASE ALL menu command. Create three new vertices and try again to construct the edges.

Why is the Edge Builder gadget so particular? If you had only created two vertices, it would have happily joined them with a single edge. If you had a dozen vertices, the Edge Builder would have had to make 66 edges to join all combinations of vertices. This might make a credible image of a bird's nest, but it would have little other use. This is why the Edge Builder will only work with either two or three vertices, building one or three edges, respectively. If, during the course of construction you had three vertices, and two of the were already joined by an edge, the Edge Builder gadget would only make the two missing edges. You can never have more than one edge between the same two vertices.



The Edge Builder Gadget At Work

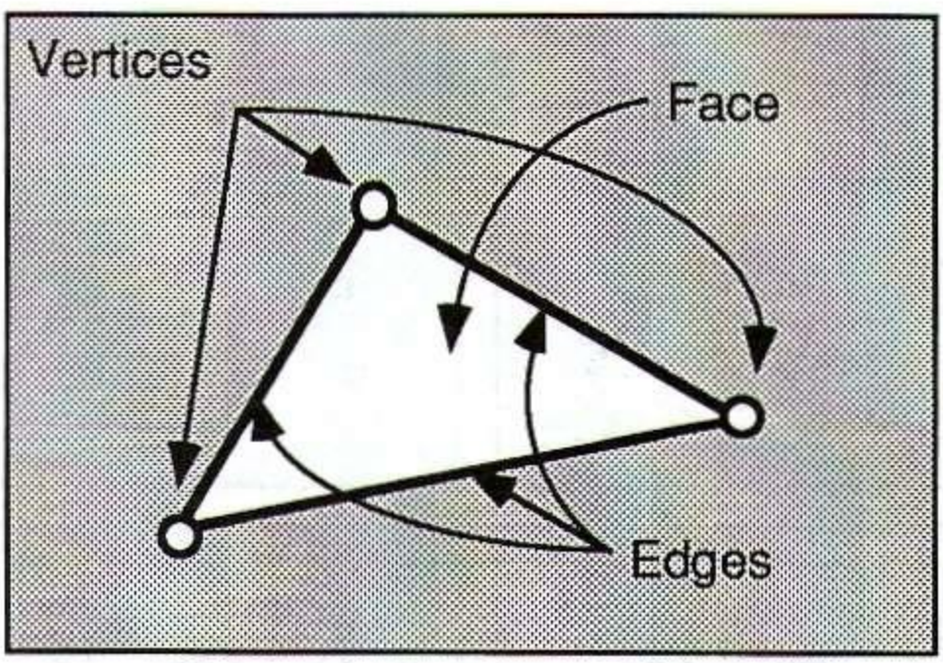
The command EDIT ADD EDGES has the same effect as clicking the Edge Builder gadget.

Faces

The next, and fortunately last, element that we must consider is called a face. In SAJr. all faces are triangular. There is no need to provide a gadget to create faces, for SAJr. has a rule:

whenever three vertices are joined by three edges, a face is automatically created.

So, unbeknown to yourself, you have already built a face. It is hardly surprising that you did not notice, because faces are not depicted in the Tri-View. They don't have to be shown, because whenever edges make a triangle, a face will be there.



Anatomy Of A Face

Although faces are not rendered in the Tri-View, they are the only parts of an object that are seen by the observer when making an image in any mode except WIRE FRAME. Vertices and edges are never seen in the final picture in SKETCH or SCANLINE mode.

To prove this, go to another window. Your triangle will probably look like a straight line in this window, because it is being viewed edge on. Move the cursor away at right angles from the line--expanding the Tri-View, if necessary--and locate the observer at this point using the OBSERVER LOCATION menu command. Next, move the cursor back to the triangle, checking in two windows to be sure of its position in three dimensions. Then use the OBSERVER TARGET menu command to point the observer at the triangle. Start the image making process with OBSERVER START.

The picture that will be created will hardly be a work of art. Indeed, even if you have done everything exactly right, you will only see the dark silhouette of a triangle. This is because we forgot to turn on a lamp. Move the cursor back to the vicinity of the observer and a small distance to one side. To create a lamp, execute the EDIT ADD LAMP menu command.

This really does seem like a lot of bother just to draw a triangle. Fortunately, we will soon be using tools that will mass produce faces so that you can sculpt shapes of any kind.

Selecting Vertices

SAJr. uses a variety of methods for choosing vertices with which to work. It distinguishes between selected and unselected vertices. As a general rule, a vertex must be selected before you can manipulate it in any way (an exception is the 'connected' group of commands to be described later). A selected vertex is portrayed as a bright yellow dot. The vertices that you created by holding down the left mouse button and clicking the right button were shown in yellow, so they were selected vertices.

To change the state of selection of a vertex, you must point at the vertex with the pointer and double click the left mouse button. Try it with one of the vertices you've made in the Tri-View. The color of the vertex will change from

yellow to a dull purple. If you double click again, it will change back to yellow. Selected vertices are yellow and unselected vertices are purple.

In order to change a vertex, the pointer must be within a few pixels of the correct spot. If SAJr. cannot find a vertex in the immediate vicinity, no vertex will be changed, but the screen will flash briefly to warn you that you missed. If this happens, move the pointer closer to the vertex and try again.

We say that double click selection works on the 'indicated vertex'. Several of the commands you will learn operate on the indicated vertex. If you want to indicate something, you do not usually have to position the cursor in all three dimensions, two dimensions suffice to make your wish known. In other words, you only need to indicate a thing in one window.

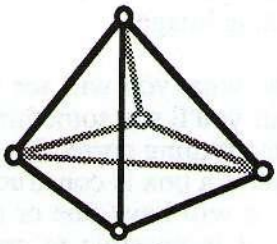
The exception to this is when two or more vertices are close to the cursor in the current window. If this happens, the vertex among them which is closest to the cursor in the other windows will be chosen. Double click selection will cause the cursor to jump to the location of the desired vertex in the other windows. Make it a habit to look at another window anyway, to see that what you wanted is what you got.

Now that you know about selection and deselection, we can tell you a little more about how the Edge Builder gadget works. You know it will construct edges between the selected vertices, provided that either two or three vertices are selected. If some other number of vertices is selected, the screen will flash a warning and nothing else will happen. If an edge already exists between two selected vertices, the Edge builder does not make a second edge in the same location..

You now have enough information to create a triangular net of vertices and edges. Turn off the selection of all but two vertices and use the Edge Builder to create a new edge. Remember that each time you complete a triangle with three

edges, a face is constructed, and it is these faces that are visible when you use the observer to generate an image.

If you move from window to window, your vertices will not all lie in one plane and your faces can construct solid shapes. Try to construct a pyramid, and then see what it looks like to the observer. Don't worry if it isn't perfect, the exercise is what's important right now.



The Edges And Vertices For A Pyramid

Remember that you have to have at least one lamp suitably placed or your constructions will just look like a dark hulk. To create a lamp, position the cursor at a suitable spot, preferably behind, above and to the side of the observer, and execute the command EDIT ADD LAMP. If you don't like the lamp where you've put it, for now just erase it with EDIT ERASE ALL LAMPS, and add a new one. You'll learn how to move lamps at the end of this chapter.

If you succeeded in fabricating a pyramid, try to build a cube. Here's a tip that can help keep it square: if you want two vertices to 'line up' squarely in the Tri-View, place your first vertex, then move the cursor to about the right place for the second vertex. In one of the other windows, the cursor will seem almost centered on the first vertex. In fact, if you go to the other window and center the cursor directly on the first vertex, you can make your second vertex and it will be right in line with the first! In this way, you can use the Tri-View as a carpenter's square. The precision is limited by the

screen, of course--Chapters 5 and 6 describe some much more powerful tools for precise placement of vertices, etc.

Back to the example at hand. Once you've laid out the cube's basic shape and connected the corners with edges, you still haven't made it a visible object. In order to give it visible faces, you will have to divide each of the square sides with an extra diagonal edge, making two triangles. If you wanted to make something that looked like an open box, you could leave out the diagonal on the top side. Try doing this, and look at the resulting image.

More than likely, what you will see is an open box, but there is a chance that you'll see something else, even if you seem to have done everything correctly. Enough suspense--it just so happens that if a box is constructed in just the right (or wrong) manner, it will have one or more 'hidden faces' inside it. You see, it is possible to create your diagonal edges (the ones that make the sides visible), so that three diagonals join at three vertices to make an extra face inside the box. This is done by selecting two adjacent vertices and one vertex diagonally opposite to them in the cube! In an orthogonal view, a view without perspective, it looks perfectly correct. This usually won't make much difference, but remember that it can happen--just in case. The next chapter will tell you how to get SAJr. to build a basic cube for you. It'll usually be best just to start with one of those and alter it to suit your purpose.

Another way to select vertices is with the Select gadget. It can be found near the top of the left border of a Tri-View window, and its icon is three dots. This gadget changes the selection state of vertices that are inside the Tri-View--only the ones visible in the Tri-View, no others. We will make use of this property in a moment.



The Select gadget looks like this.

Here's how the gadget works: if one or more vertices in the Tri-View are selected, then the gadget deselects the vertices. If no vertices in the Tri-View are selected, then the

gadget causes all of them to be selected. You can use the gadget to toggle all the visible vertices between selected and deselected.

A useful trick, when you wish to select only some vertices of an object, is to move, expand, or contract the Tri-View until only those vertices that you wish to change remain inside. The Select gadget will then change only these vertices. This is an excellent way to select one side of a cube, the end of a wing, the front or back of some letters, etc.

There are still more ways to change the selection state. The menu command EDIT SELECT ALL and EDIT DESELECT ALL can be used to select or deselect every vertex in the scene, not just those in the Tri-View. The commands EDIT SELECT CONNECTED and EDIT DESELECT CONNECTED operate on all vertices that are connected to the indicated vertex. Any other vertex that is linked by an edge, either directly or indirectly through a network of edges, to the indicated vertex is considered to be connected to it.

As an example, if you were to reload the little house and try out the SELECT and DESELECT CONNECTED commands, you would find that the faces that compose the roof were connected together, but were not connected to the faces in the walls. All four walls are connected together. This makes it easy to select the roof, without changing the walls. Later in this chapter we will see how to literally raise the roof.

Another useful command for selecting vertices is EDIT SELECT SWAP. As its name suggests, all selected vertices become unselected and all unselected vertices become selected. This command works on all vertices, not just those that lie inside the Tri-View.

A single vertex can be selected or deselected with the commands EDIT SELECT INDICATED VERTEX and EDIT DESELECT INDICATED VERTEX. In either case

you must indicate the vertex that you intend to change by placing the cursor close enough to it.

The final selection commands that we will consider at this time are EDIT SELECT INDICATED EDGE and EDIT DESELECT INDICATED EDGE. If you place the cursor on an edge, these commands will select or deselect the two vertices at the ends of the edge. This can be useful when dealing with very complicated objects when several vertices are clustered together and you do not know which one is connected to a particular edge.

There are so many ways of selecting vertices (a few more will be added in other chapters) because the key to manipulating complicated objects is to be able to select just those vertices that you need for a particular operation.

Erasing Objects

SAJr. is an editor for solid objects. So far, we have seen how to create vertices and edges, and--by the rules--make faces. What happens if a mistake is made?

The EDIT ERASE portion of the menu provides a number of options. These commands remove items from the current scene in RAM memory, but they have no effect on scenes that have been saved, unless you save over an old scene with the altered one. Be a little cautious with the erase commands, anything you erase is gone forever--unless you have saved the scene beforehand. It's a good idea to save your intermediate work at intervals.

The first selection in the ERASE submenu, EDIT ERASE SELECTED VERTICES does what you might expect. Only selected (i.e., yellow) vertices are removed. Since an edge is defined in terms of vertices, any edge that is connected to a selected vertex is also deleted, and since a face requires three edges, the removal of edges may result in faces being erased also. When you remove an object's vertices, you remove an object's foundation.

Note that EDIT ERASE SELECTED VERTICES works on all selected vertices, so you should be careful that you do not have any selected vertex outside the Tri-View, otherwise it will vanish also. It's a good idea to use EDIT DESELECT ALL before selecting vertices to erase, or at least check the status display (Chapter 3) to make sure none are already selected.

The next erase command, EDIT ERASE SELECTED EDGES, is very similar. It operates on selected edges, without removing any vertices. A selected edge is one that has selected vertices at both ends. For example, if you had made a cube with diagonal braces on all six sides, you could open up one side by removing a brace. Simply select the vertices at either end of the edge to be removed. The EDIT ERASE SELECTED EDGES command will remove the edge. Again, if any edge of a face is removed, the face disappears.

In addition to being able to erase selected items, you can also erase them individually. The commands EDIT ERASE INDICATED VERTEX and EDIT ERASE INDICATED EDGE work on a single edge that is indicated by the cursor. Again, if more than one edge is near the cursor, look in another window to make sure the correct one is indicated.

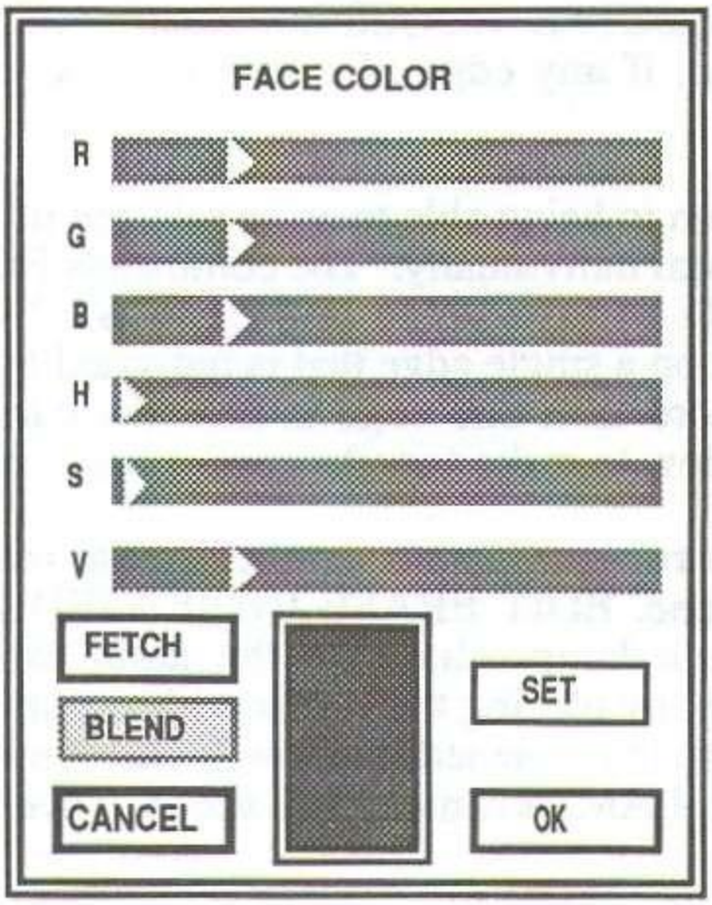
The next two commands can be used to remove lamps from the scene. EDIT ERASE INDICATED LAMP will erase a single lamp. Indicate the lamp as you would anything else, by placing the cursor within a pixel or two of it in the currently selected window. The command EDIT ERASE ALL LAMPS is much less subtle, it just removes all the lamps.

The last erase command that we will consider at this time is EDIT ERASE ALL. It removes all vertices, edges and faces as well as all lamps. Because this could be catastrophic--for example if you have just built an elaborate scene and neglected to save any part of it to disk (which is not wise in any case)--SAJr. will ask if you really want everything erased. This gives you a chance to retract the command.

Color

By now, you must have become aware that the objects you have made have been white, and that the effects of shading have only created different shades of gray. Every time SAJr. creates a face, the face is given the current default color, whatever that happens to be. White is the default color when you run SAJr. If, at any time, you want to see or change the current default color, you need to execute the menu command EDIT MODIFY FACES.

The requester that pops up can be used for several purposes, as we shall see.



The Modify Faces Requester

The current color is displayed in the box near the bottom of the requester. The set of six sliders can be used to control the color--ignore everything else for now. The first three sliders control the amount of red, green and blue. To

operate a slider, point at the triangular knob and hold down the left mouse button. If you move the mouse from side to side, the knob will move to follow it, and the current color will change. With a little practice, you can mix the red, green and blue to achieve almost any hue.

If you find it difficult to get just the right shade with the RGB sliders, try using the lower three that are labelled H, S and V for hue, saturation and value. It is a good technique to start by moving the S slider and the V slider all the way to the right. If you then move the H slider very slowly it will change the color smoothly from red to green to blue and back to red again, with many shades in between.

Having established the correct hue, you can move the saturation or S slider to the left in order to dilute the color with white. Finally you can use the V slider to make the color darker or brighter.

Once you have found the correct color, click the gadget marked 'OK' at the bottom of the requester. From now on, any new faces that are generated will have that color.

The color of the faces can also be changed at any time. First, you must select the faces whose color you wish to change by selecting the vertices that compose the faces. Then execute the command EDIT MODIFY FACES and use the sliders to select the right color. To set the color of the selected faces, click the the button marked SET. The change will take place when you click OK.

You will have to exercise some care in selecting faces to be re-colored. Consider the case of a red triangular face in the middle of a blue surface. The red face's vertices are all shared with blue faces. Suppose you wish to change the color of all the blue faces from one shade of blue to another shade of blue. If you select all the blue faces, you will also select the red face! This may not be what you wanted. If this sort of situation occurs, proceed as follows:

1) Select all the vertices of the blue faces, this will also select the three vertices of the red face.

2) Deselect one of the vertices of the red face and modify the color of the selected face. The red face will be unchanged, but some of the blue faces will be unchanged also.

3) Select the vertex of the red face and deselect the second vertex of the red face. Modify the face colors again.

4) Repeat step 3 for the last vertex of the red face. Finally all the blue faces are changed, but the red face is unchanged.

The MODIFY FACES requester can also be used to examine the color of one or more faces. Select the faces that you are interested in by selecting the vertices that compose them, then invoke the modify faces requester.

If all the selected faces have the same color, then the button marked FETCH COLOR will be solid and the button marked BLEND will be shown in 'ghosted' text. If the selected faces do not all have the same color, then the FETCH button will be ghosted and BLEND will be solid.

When the FETCH button is active, you can click on it to make the default color the same as that of the selected faces. The BLEND button is similar, except that it will display the average of the colors of the selected faces. Be warned that the average of several colors is not always what you might expect, it is defined by averaging the red, green and blue components separately.

Again, the displayed color will become the current color when you click OK. If you click SET in the FACE COLOR region and then click OK, the color of any selected faces is also changed.

If you call up MODIFY FACES and both the FETCH and BLEND buttons are ghosted in the FACE COLOR region, it means that no faces are selected. For a face to be selected, the vertices at all three corners of the triangle must be selected. It is easy to have many vertices selected without

having any faces selected. One particularly confusing situation can arise if two vertices have been created in the same place. This can cause something that looks like a triangle to exist, and still not be a face.

The Grabber

Up to this point, we have assumed that the objects that you have built are perfect, that you have never made a mistake or changed your mind. Except for erasing all or part of an object and rebuilding it, there has been no way to alter the objects you have created.

The Grabber is a low level tool that you can use to change the position of one or more vertices. Like most SAJr. facilities, the Grabber manipulates selected vertices, and leaves unselected vertices alone.

You invoke the Grabber tool by clicking the Grabber gadget, which resides in the lower left corner of each Tri-View window. Its icon is supposed to look like the crane operated gripping tool that is used in automobile wrecking yards.



The Grabber gadget looks like this.

If you click on the Grabber icon with the left mouse button, you will see its jaws close. It is then active. While the Grabber is active, the normal crosshair cursor will be replaced by a copy of the Grabber gadget icon.

The Grabber may be dropped (turned off) by clicking on its gadget a second time, or by clicking the right mouse button. As an alternative to the gadget, the Grabber tool may be activated by the menu command TOOLS GRABBER.

While the Grabber is active, you may move the new cursor anywhere in the scene, just like the normal cursor. The difference is that any vertices that are selected will

follow along, maintaining their position relative to the cursor.

When you move a vertex, you also move or distort any edges and faces that are built upon it. For instance, if you selected and grabbed just one vertex of the cube you built a short while ago, you could move it without moving the rest of the cube. The edges and faces built upon the moved vertex would stretch to accommodate its new position. You could also select and grab the entire cube, then move it without any distortion.

Now give the grabber a try. First clear the Tri-View with an EDIT ERASE ALL command. Next make three vertices, and join them with a click on the Edge Builder gadget. Deselect one of the vertices and turn on the Grabber. All the time that the Grabber is on, the selected vertices will move whenever the cursor is moved. The edges will stretch to follow. Notice that the cursor doesn't have to be in any particular place when you invoke the Grabber. The selected vertices will always maintain the same position relative to it when the Grabber is active.

To try a more elaborate example, EDIT ERASE ALL to clear the Tri-View. Then PROJECT LOAD SCENE and request LITTLEHOUSE. When it has loaded, deselect all the vertices with the EDIT DESELECT ALL command and move the cursor to one of the vertices at the top of the roof. Then execute the EDIT SELECT CONNECTED command and the whole roof structure will be selected. Turn on the Grabber and move the mouse pointer to the North window (the one with W, U, E and D at its edges). Then move the cursor up and raise the roof!

The grabber can also be used to move a lamp. To do this, you must first DESELECT ALL, if there are any selected vertices, otherwise they would be grabbed. Next, indicate a lamp with the cursor. When the grabber is turned on, it can then be used to drag the lamp to a new location. Chapter 7 also describes this.

The Expand, Contract, Center, and Move Tri-View gadgets all work correctly when the Grabber is active, so you can move objects and lamps anywhere in the scene. You must press the right mouse button to use menu commands, so these are not available when the Grabber is active. Avoid the Rotate (Chapter 6) and Select/Deselect gadgets when the Grabber is active.

Techniques For Using The Grabber

The Grabber has some extra flexibility since the cursor doesn't have to be placed directly on the selected vertices that will moved. One thing you will immediately notice is that the cursor graphic can get in the way and obscure area of the Tri-View in which you are working! The easy way to get around this is to stop thinking of the cursor as the sole reference point for placing the selected vertices. After you select the vertices, set the cursor off to the side of the Tri-View, and then click the Grabber gadget. This offset cursor seems strange to work with at first but allows you to see more easily what is happening.

Another technique of using an offset cursor is to place the cursor at the very edge of the Tri-View window before activating the Grabber. This allows you to use the edge of the window as a straight edge to move the object precisely in one of the cardinal directions. This is done by picking up the Grabber and pulling the mouse pointer just outside the Tri-View window. The Grabber cursor will stop at the border. Likewise, pushing the Grabber cursor with the window border using the Move Tri-View gadgets will force the selected vertices to move in a line at right angles to the window edge.

Perhaps the best feature of the Grabber is that its last move can be undone. Clicking on the Select gadget before turning off the Grabber will cause the selected vertices to become deselected and snap back to their original positions, but make sure that no selected vertices are outside the Tri-View, or the Grabber's will only be partially undone.

Notes

CHAPTER 5

POWER TOOLS

In this chapter we will present some of the 'power tools' that enable you to easily build elaborately shaped objects. The objects you create using these tools are made up of the same elements that were introduced in the last chapter--vertices, edges, and faces--so you will be able to adjust and manipulate these new structures in the same way, if necessary.

Spheres

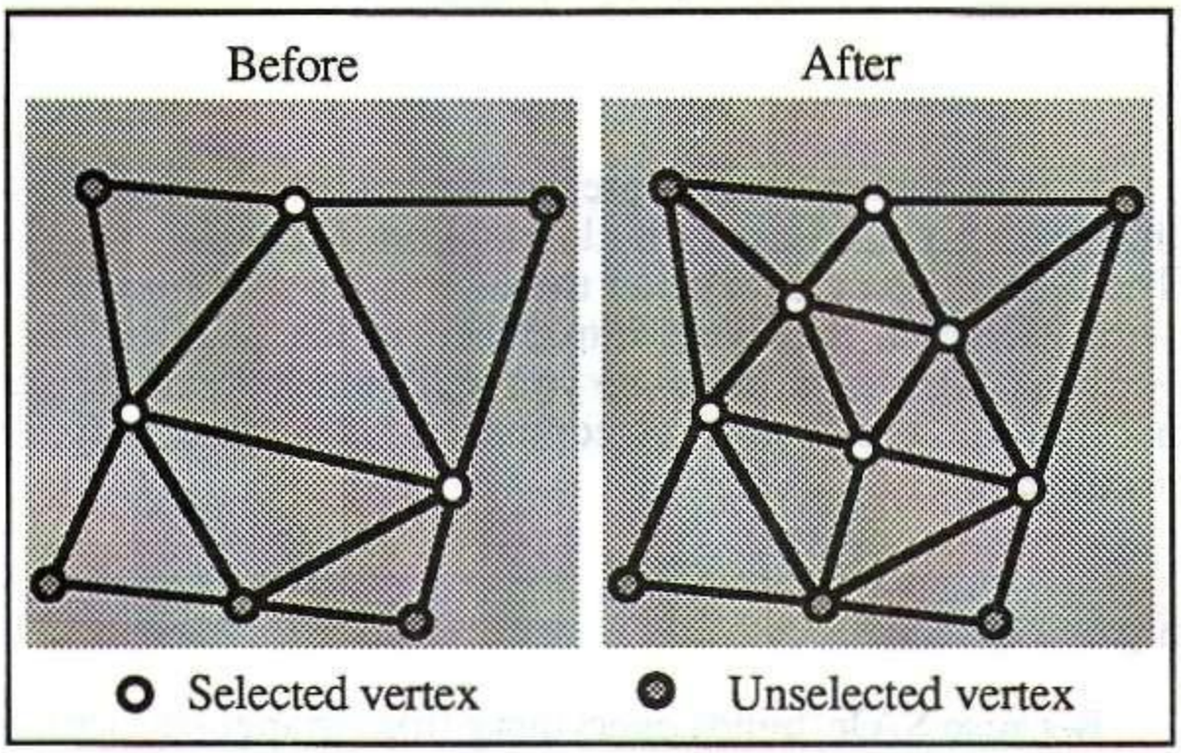
Because SAJr. builds everything from triangular faces, it cannot represent a sphere exactly. It so happens, however, that a geodesic structure--constructed of triangular elements--can represent a good approximation of a sphere. Though it would be possible to build such a structure using the tools discussed so far, it would take extreme patience and an unusual fondness for geometry.

Fortunately, SAJr. knows how to construct spheres, and other objects, for you. To have it build a sphere, execute the menu command EDIT ADD SPHERE. A requester will pop up and ask you how many times you wish to subdivide the structure. For now, simply press the OK button and SAJr. will construct a 20-faced geometrical figure called an icosahedron.

To take a look at this new object, set up a lamp and make a picture of this object with the SCANLINE mode. It is a pretty poor approximation to a sphere

To make a better approximation of a sphere, we may subdivide each face. The EDIT DO SUBDIVIDE menu command will divide each selected edge in two, creating a

new vertex at the midpoint of each edge. Then these midpoints are linked with new edges.



The Subdivide Operation

Each selected face will be divided to make four. Try this operation on the icosahedron, being sure that each vertex is selected.

The resulting object will have 80 faces, but will still look like an icosahedron, since the new faces lie in the same planes as the old ones. What we need to do is to 'inflate' the object by pushing the new vertices outwards a small amount.

You might have already guessed that you can subdivide the sphere automatically by specifying the number of subdivisions when the requester for EDIT ADD SPHERE pops up. Instead of just pressing the OK button, click on the box containing the default subdivision number, erase the default with the DEL or BACKSPACE keys, then type in the desired value and hit RETURN or click OK.

Subdividing Objects

The preceding section described how the command EDIT DO SUBDIVIDE may be used to increase the detail of a sphere by dividing each selected edge into two and each selected face into four.

The command may be used on other objects as well, but it's important to remember that, by itself, EDIT DO SUBDIVIDE only adds vertices and edges--not detail. Never subdivide an object unless you intend to move or alter the added vertices in some way.

For instance, if you wanted to add a nose to a potato, there might not be enough vertices in the area of interest to construct this detail. You could select just this part of the surface and subdivide it until there were enough vertices to work with. Then you would have to move these vertices into the shape of a nose. Of course, you could also construct the nose separately and simply move it into place--there are often a number of ways to achieve a desired effect.

There is one special function that the SUBDIVIDE command will perform. If only two vertices are selected, and they are not joined by an edge, then the SUBDIVIDE command works quite differently. A requester pops up asking you how many vertices you want inserted. New vertices are placed evenly between the two selected ones and they are joined by edges that lie along a straight line. No faces are created, so this new object would not be visible in an image.

Other Curved Objects

Besides spheres, the following curved objects can be created with the appropriate EDIT ADD commands: ADD HEMISPHERE, ADD DISK, ADD CIRCLE, ADD CYLINDER ADD TUBE and ADD CONE adds hemisphere, disk, circle, cylinder, tube and cone, respectively. In each case you will have to enter a number to specify the degree of detail that is required.

In each case except will the circle and tube, a complete closed object will be built. For hemispheres you must specify a number corresponding to a number of subdivision operations, while for the other ADD commands, the number that you give represents the number of rotation steps for a full circle.

Cubes

Any angular shape, like a cube, can be represented precisely by SAJr. However, each rectangular face must be divided into triangles by a diagonal edge. The EDIT ADD CUBE command will construct a perfect cube in the Tri-View--complete, with a diagonal division on each side.

The cube that is constructed will be automatically sized to fit comfortably within the current size of the Tri-View. All of the EDIT ADD sizing commands behave this way. In the next chapter we will introduce a mechanism called EXPAND that can change the size of objects, as well as their proportions.

With the Expand mechanism, a simple cube may be transformed into any size or shape of box. For now, though, we will use the Grabber gadget to do a similar transformation. Although they are somewhat more awkward than EXPAND, the techniques discussed here can be used to create a number of more subtle effects.

Suppose we wanted to build a block that is twice as large in one direction as it is in other directions. First, clear the Tri-View with an EDIT ERASE ALL. Then make a basic cube with EDIT ADD CUBE.

Since the cube is sized to fit the Tri-View, we need to expand the Tri-View to make room for our stretching. Click on the Tri-View Expand Gadget twice. Now we want to select the four vertices that make up one end of the cube. One way is to use the Move Tri-View gadgets to put that part of the cube outside the Tri-View. Then use the Select

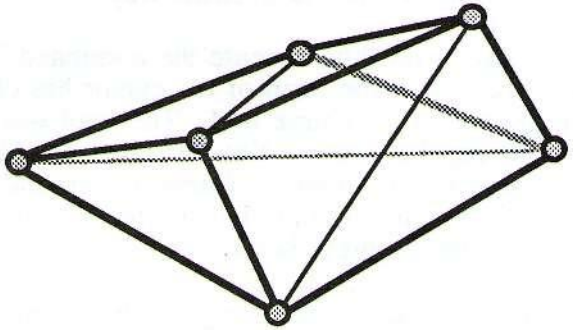
Gadget to select the four vertices that remain. When this is done, Contract the Tri-View so you can see the whole cube.

Now look at the three windows and find one that will let you move the cube's selected side away from the unselected side. The window whose Move Tri-View gadgets you were using should do. Finally, turn on the Grabber and--in the window you just chose--pull out the end of the cube until it reaches the desired length.

If you do not have a steady hand, you may find that you have pulled the cube out to a lozenge shape. Try to remember this shape--and any other shape you create by accident. You never know when you might find a use for it!

Prisms

The command EDIT ADD PRISM works just like the command for cubes. The prism will be oriented as though it were the roof of a building, but in the next chapter you will see how to tilt and rotate it into any position.



Prism

Just as with the cube, you can easily alter this basic shape to achieve any number of effects. For instance, if you

want to create a steeply-pitched roof, you can select the two upper vertices by double-clicking them and use the Grabber to raise the ridge. You can also move the ridge side to side, creating interesting asymmetrical effects. Similarly, if you want anything other than a square plan, turn off the Grabber and select just one end of the roof, then grab and move that.

Again, subsequent chapters will introduce even quicker ways to alter basic shapes like the prism.

Drawing Curves

So far, the shapes we have constructed have had simple, regular profiles either circular, square, or triangular. The majority of objects in the real world are not so regular, however. Most are clearly distinguishable by a unique profile.

SAJr. has several features which allow you to take one or more profile curve and create solid objects from them. In SAJr., these curves are approximated by a series of vertices joined by edges. Although we could make them using the methods of Chapter 4, there is an easier way.

To see how to do this, execute the command TOOLS CURVE. Notice that the form of the cursor has changed: you have picked up the Curve tool. This tool works very much like the regular cursor. You can click or hold down the left mouse button to move the Curve tool around. If you press the right button while the left button is held down, a new vertex is created--just as before.

Unlike the regular cursor, though, subsequent vertices will be joined to their predecessor with an edge. This lets you easily create a linked set of vertices. As you move the Curve tool, the last vertex will be linked to the cursor by a dotted blue "rubber band line". This reminds you that you are about to create an edge, and shows you the orientation of the edge.

While the Curve tool is active, the gadgets for moving and sizing the Tri-View may still be used to move around the scene.

You may release the Curve tool at any time by clicking the right mouse button. If you position the tool very close to an existing vertex, the curve will be closed by creating an edge to the vertex, and the Curve tool will be dropped. In this way you can create closed loops.

The Edge Maker Tool

A closed loop would be just what you needed if you had to create a flat object with an arbitrary shape, like a letter on a billboard. If you tried to make an image of such an object, hoawever, it would be invisible, because although you would have created vertices and edges, there would be no triangular faces. You have to add interior edges in order to triangulate (produce the triangular faces) the interior of the loop. Usually there are many different ways to do this but rendering images will be faster if you avoid long skinny triangles.

There is a tool that makes the job of triangulating the interior of a loop very easy. To try it, first make a closed loop--it doesn't matter if the vertices are selected or not. Next, call the command TOOLS EDGE MAKER. Notice that the cursor changes to a shape that may remind you of the Edge Builder Gadget.

To use this new tool, move the pointer to a vertex, then hold down the left mouse button and click the right button once. This starts the edge construction. Move the cursor and notice that a rubber band line goes from the vertex to the cursor. Move the cursor to a second vertex and do the same thing again--hold left, click right. An edge will be made between the two vertices. Repeat the process until you have completed the triangulation.

Spinning New Objects

Many objects have symmetry about an axis. Examples include a doughnut, a spindle, a chess pawn, a candlestick, and an hourglass. All of these objects can be described as a curve which has been swept about an axis. The EDIT DO SPIN command is the mechanism that SAJr. provides for generating these shapes.

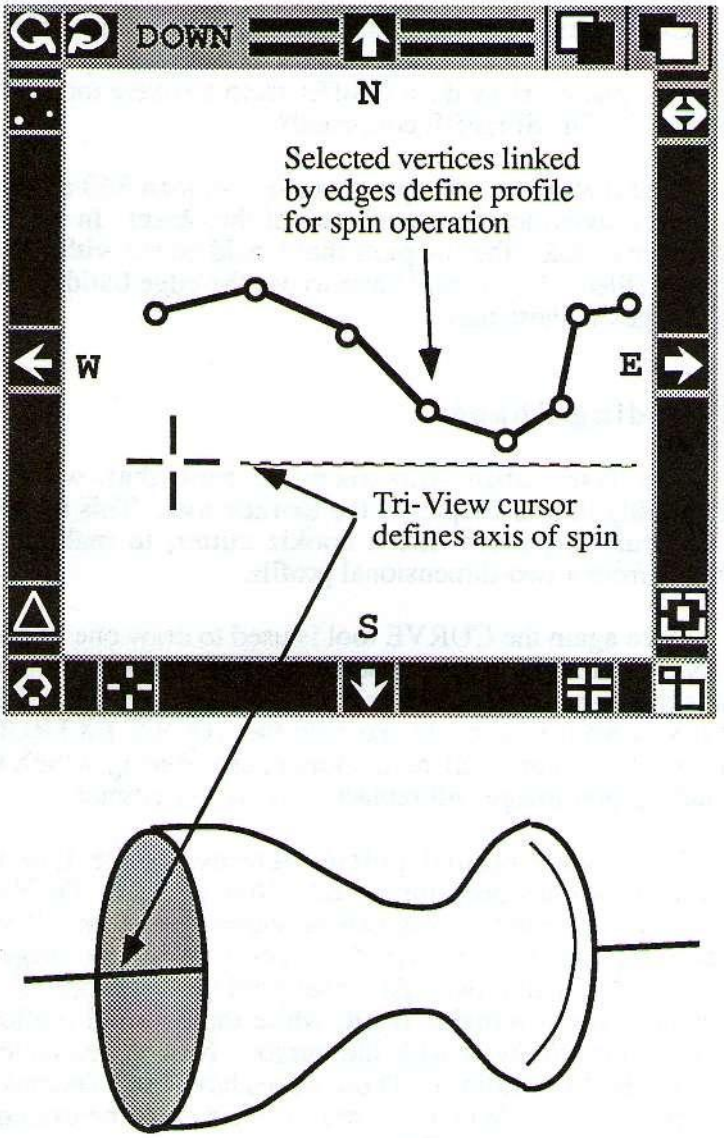
To make this command work, you must first create a curve that describes the outline of your object. This is simple enough if you use the tools that have already been described.

Once you have designed your pattern, make sure no other vertices are selected in the scene (with EDIT DESELECT ALL), then select the pattern curve with EDIT SELECT CONNECTED.

Next you must indicate the axis about which the pattern is to be spun. Do this by placing the cursor some distance above or below the pattern curve in one of the Tri-View windows. It is assumed that the axis runs horizontally across the screen, passing through the cursor in the selected window.

When you are satisfied with the curve and the location of the axis, execute the EDIT DO SPIN menu command to generate a new shape. The command will display a requester asking how many steps are to be taken as the curve is swept around the axis. You must use at least three. More steps will create a smoother object. But more steps will also consume more memory, having an effect similar to using many subdivisions on a sphere. The default value of 12 is usually adequate.

After you have selected the number of steps, you will be asked to provide a spin angle. Normally you should accept the default angle of 360 degrees, in order to produce a full surface of revolution. If you decide to use an angle less than 360 degrees, then a partial spin will be performed--just the thing to model a slice of cake.



The Spin Operation

Experiment with the spin tool to see how many different shapes you can build. Try using a semicircular pattern to make a sphere. How does it differ from a sphere made with the EDIT ADD SPHERE command?

Note that when you spin a curve less than 360 degrees, you only construct the curved part of the object. In the case of a slice of cake, the flat parts that would be cut with a knife are not filled. You would have to use the edge building tool to triangulate these parts

Extruding Shapes

The last outline-transforming tool that we'll be examining in this chapter is the extrude tool. This tool lets use a curved pattern like a cookie cutter, to make solid shapes from a two-dimensional profile.

Once again the CURVE tool is used to draw one or more curves or closed loops with selected vertices. When you are satisfied that you have created the cross section of the shape that you want to extrude, execute the TOOLS EXTRUDE command. There will be a short pause, during which the grinding gear image will replace your mouse pointer.

When your normal pointer returns, notice that the Grabber tool has been turned on. Now select a Tri-View window where the curves can be viewed edge-on. If you move the cursor in this window, you will see the original patterns divide into two. An unselected (purple) copy of the original pattern remains fixed, while the selected (yellow) copy can be dragged with the cursor. New edges appear, joining the two outlines. These edges have been created by the extrude tool. When you are satisfied with the extruded shape's form, click the Grabber icon to turn it off.

If you had traced out the outline of a building in plan form, you could use the extrude tool to raise the walls with one stroke of the mouse. It can also be used to build three dimensional alphabetic characters, any number of stamped or extruded forms, and many other objects.

One useful property of the extrude tool is that if you use it on a curve that has been triangulated, the interior edges do not extrude to form interior partitions. This prevents useless faces from being created within extruded objects, where they would waste memory and processor time.

The extrude tool is ideal for making three dimensional alphabetic characters. Generate the outline of the character using the Curve tool and triangulate it. The outline can then be extruded into a solid character. If you only plan to look at the front of the character, you can leave the triangulation step until after the extrusion and only triangulate the front of the character. This will result in a simpler object that will use less memory and will probably require less time to render. You can easily assemble an entire set of characters in this way.

NOTES

CHAPTER 6

MORE TOOLS

In this chapter we will look at more ways that we can manipulate and build objects.

Saving And Loading Objects

If you build an object, then you can save it to disk with with the PROJECT SAVE SCENE command. A file requester will pop up to enable you to specify the name that you wish to use for the file. This requester was described in detail in Chapter 2. The requester lets you move around among the volumes and drawers in your Amiga's file system. When you have found the right drawer, you may either select a pre-existing filename or type in a new one in the provided box. If the name that you pick for saving the scene is one that has already been used, the old file will be destroyed. You will be warned before this happens and allowed to cancel the command.

A scene is made up of four parts: lamps, the observer, the world, and objects. When you have finished picking a filename for the scene, another requester will appear which lists these parts. Next to each part's entry is a 'toggle gadget' which may be switched from YES to NO and back just by clicking within its border. Items marked 'YES' will be saved, items marked 'NO' will not.

'Lamps' is the top item in the list. It specifies whether lamps will be saved. You may not want to save the lamps in a scene if you are only saving objects for use in another scene. On the other hand, you might have a particularly pleasing arrangement of lamps that you wish to save for other scenes.

'Observer' is the next item on the requester. If the Observer gadget is set to NO, the observer's location and target will not be saved. Neither will any of the parameters associated with the observer menu, such as rendering mode and image size and type. These parameters are saved with a scene in the Observer part.

The 'World' is described in more detail in Chapter 7. It includes details about the sky, ground, and background illumination for a scene.

'Objects' is the final and most important entry on the requester. The objects in a scene are its vertices, edges and faces. The color of each face is stored in this part of the file. If this gadget is set to 'NO', no objects will be saved at all. If the gadget is set to 'YES', they will be saved according to the gadget marked 'Which?' at the bottom of this requester.

The 'Which?' gadget is usually set to 'ALL', which means that all objects will be saved if 'Objects' is set to YES. If the gadget is set to 'SELECTED', however, only those objects in the scene that are selected (with yellow vertices) will be saved.

By carefully selecting which parts of a scene to save or load, you can combine the best parts of different scenes--lamps from one, objects from another, etc. Be careful, though! If you leave some of these items set to 'NO', then the next time you save a scene, you might not save all of the parts of the scene that you wanted to.

PROJECT LOAD SCENE behaves very much like PROJECT SAVE SCENE, except, of course, that it loads the scene data back into the computer. LOAD SCENE also allows you to choose which parts of the scene to load. The selection states of objects are not stored, so all objects that were saved in a scene file will be loaded in and you may choose whether the objects that are loaded will be selected or not.

The LOAD SCENE command does not erase any objects that are already present in the SAJr. universe. This feature

allows you to load in different parts from different scenes, as we have seen. If you want to clear the slate and start over with a scene you've loaded, do an EDIT ERASE ALL before re-loading the scene.

If you only need to save or load objects, then you may use the commands PROJECT SAVE OBJECT or PROJECT LOAD OBJECT. These operate a little differently from the commands that work with complete scenes. You are not given the choice of saving or loading lamps, the observer, or the world, and only the selected objects are saved.

There's a more subtle difference as well. LOAD SCENE will always load objects into the same 'absolute' position in the SAJr. universe where they were when SAVE SCENE stored them. On the other hand, SAVE OBJECT records the position of the saved vertices relative to the cursor. When the objects are loaded in by LOAD OBJECT, they will be placed the same position relative to the cursor's location, no matter where that position is in the scene.

Suppose that you had constructed a model of a grand piano and all the vertices were selected. Then you could position the cursor under one leg and save the object to a disk file. If, at another time, you wished to introduce a grand piano into the Tri-View, you could move the cursor to the position where you wanted to position the leg of the piano and execute the PROJECT LOAD OBJECT command. The piano would now be correctly located in your scene.

If you need numerous copies of the same object, you can also use the command PROJECT LOAD OBJECT to add multiple copies.

Rotating Objects

We may not always want our objects to be lined up with the axes of the Tri-View. If we are building a model of a city, for instance, and we want to use the EDIT ADD CUBE command to produce buildings, we have yet to see an easy

way to produce a cube that faces to the southeast, or any other 'off axis' direction.



The Rotate gadgets

The Rotate gadgets provide a way to change the direction our objects are facing. The gadgets may be used to rotate any selected object. To see how this gadget work, do an ERASE ALL and make a cube. Position the cursor at the center of the cube and select all eight of the vertices (the Tri-View's Select gadget should do nicely for this). If you click the left mouse button on one of the Rotate Gadgets on the top left side of the Tri-View window, you will see the cube rotate 5 degrees for each click. If you hold down the right Amiga key as you click, each rotation step is increased to 45 degrees. Similarly, if the left Amiga key is held down, the rotation step is only 1 degree.

The left and right ALT keys can be used to modify the rotation gadget even more. If the gadget is used with the left ALT key, the rotation is only 0.1 degrees. The right ALT key increases the rotation angle to 180 degrees, so this is a quick way to turn an object about-face.

If you move to another window, you will be able to revolve the cube around a different axis.

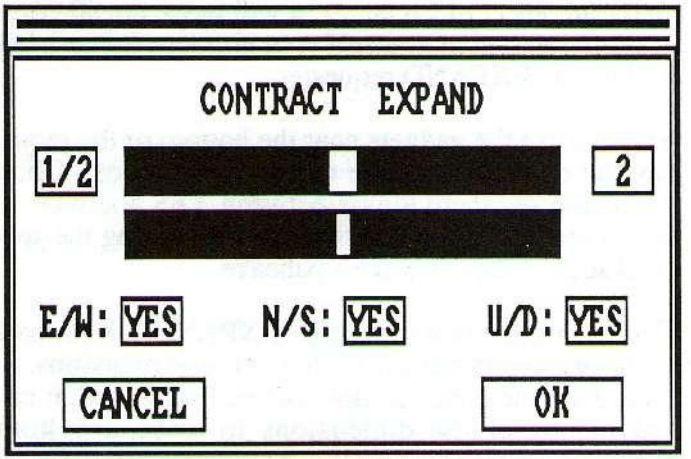
To see a different use for the rotate gadgets, expand the Tri-View to about ten times its original size and position the cursor at some distance from the diminutive cube. Click the Rotate Gadget and you will see the little cube move in a circular orbit around the cursor.

In summary, the Rotate Gadget rotates all the selected vertices around the cursor. Be careful that you do not have any vertices accidentally selected outside the Tri-View because they will be rotated also.

For now, you will probably want to select all the connected vertices of an object to be rotated. Otherwise, only part of the object will be rotated and a tangled mess can result. As you begin to create more detailed objects, you may find it convenient to select portions of an object's surface and alter its shape by clever manipulation of the cursor and rotate gadgets.

Expansion And Contraction

Each of the EDIT ADD sequence of commands creates an object just a little smaller than the size of the Tri-View. This trait would be inconvenient if we wanted an object of a specific size as we could not specify the size of the Tri-View exactly. We could only change the object size in steps. Besides, if we found that we had created an object of the wrong size, our only recourse would be to erase it and try again, or to start fiddling around with the Grabber. That would hardly be practical for a pseudo-sphere with 320 faces. Fortunately, there is a command that enables you to change the size of an object with great precision.



The Expand Requester

Let's learn by doing. First, make a one-subdivision sphere and select all its vertices. Move the cursor so that it is near the center of the sphere in all three Tri-View windows. Then, execute an EDIT DO EXPAND command. An expansion control requester will pop up. Unlike some other requesters, this one has a drag bar along the top. Move the requester so that you have a good view of each of the Tri-View windows.

The EXPAND requester has two slider gadgets across its width. You can use the upper slider to continuously change the size of the selected object. Moving the slider to the left makes the object smaller and moving the slider to the right makes the object larger. This movement of the slider moves the selected vertices towards or away from the cursor's location in a proportional fashion, so that the object grows smaller or larger. The lower slider operates just the same way, except that it is a much more precise control. You will hardly be able to see the change it produces.

There are two other sizing gadgets, one at either end of the sliders. One is marked '1/2' and the other is marked '2'. As you might guess, you can click these gadgets to make the selected object either half- or double-sized. Notice that if you make an object much larger, it will move outside the Tri-View. Be sure to give yourself enough room to work before you call up the EXPAND requester.

You can use the gadgets near the bottom of the requester to limit the expansion to one or two dimensions. Click on the boxes and see them toggle between YES and NO. Turn off two of the directions and then try contracting the sphere. You will see it flatten out like a pancake.

This is an important principle. EXPAND can be made to work in any combination of the three dimensions. For instance, a simple cube can not only be lengthened, it can be resized in any of the dimensions to make any kind of rectangular block. The compass markings on the Tri-View windows can be helpful in picking which dimensions you want to resize.

Even more variety can be achieved by only selecting portions of an object to be resized. By selecting the top four vertices of a cube and shrinking them, you can create a trapezoidal pedestal (a pyramid with its top cut off). You can select individual vertices of a sphere and resize them, creating a number of spikes or dimples on the sphere's surface. Starting with a few simple objects and some imagination, you can quickly create any number of shapes.

Note that the expansion is relative to the cursor. If the cursor is not at the center of an object, the expansion controls will make the object move, as well as change shape.

Manipulating Shapes

If you were to EXPAND a sphere along a one dimensional axis, you might expect that it would lengthen out into an ellipsoid shape (oval solid). This is indeed the case. The Grabber Tool can also be used to perform an EXPAND function of sorts, by selectively lengthening middle sections of an object. This distinction between expansion methods will come in handy over time as you build more diverse shapes.

Suppose that you wish to create a long cylinder with spherical ends. First, construct a sphere with the EDIT ADD SPHERE command. Use a subdivision value of one or two. Then, move the Tri-View so that exactly half of the sphere is sticking out of the Tri-View. Now, if you use the Select Gadget, you will find that you have selected all the vertices on one hemisphere. Click the Expand Tri-View gadget to make room for what will follow. Turn on the Grabber and you will find that you can pull the two halves of the sphere to make the cylinder with rounded ends.

With tools such as EXPAND and the Grabber, you can build a great variety of objects just by altering pre-existing ones.

Duplicating Objects

Sometimes you will need several copies of the same object. The command EDIT ADD DUPLICATE, will cause every selected vertex, edge and face to be duplicated. The duplicate is produced at exactly the same location as the original, so it won't be visible at first. If you use the Grabber to move the original version (since it is selected), then the copy will be revealed. The copy is created in an unselected state. You can use this command, like a brush in a paint program, to scatter duplicate objects wherever you please.

Of course, every additional copy of an object puts an additional load upon memory and computation. If you plan to create several copies of an object, you will probably want to make your prototype with as few vertices as possible.

Building Mirror Images

Some objects, like a gloves, come in pairs. As in the case of a pair of gloves, paired objects are often mirror images of each other. If you had painstakingly built a model of a left glove, you wouldn't want to build its mate by hand and you don't have to. SAJr.'s REFLECT command will build the mirror image of an object automatically.

To use this facility, you must select all the vertices of the object to be reflected. Choose one window and position the cursor in it to the left of all the selected vertices. The imaginary plane of reflection will be vertical in the current window and it will pass through the cursor position. Execute the command EDIT DO REFLECT and a mirror image will appear to the left of the cursor.

You may place the cursor, and hence the plane of reflection, at any position in the window, but if a selected vertex lies on the plane of reflection (or within a few pixels of it), then that vertex will not be duplicated. Rather, that vertex will be shared between the object and its refection.

This procedure provides an easy way to construct closed objects that have a plane of symmetry.

For example, a human head is approximately symmetrical. If you were to build the left half of a head, you would only need to construct one eye, one ear and half a nose and mouth. You would have to take care that all the vertices that are supposed to lie on the center plane do so exactly. To check this alignment, see that the vertices lie in a straight line in one of the windows. If you select this window and position the cursor on this line, then the reflect operation will construct a complete head with the halves 'stitched' together.

Multiple Use Of The Extrude Tool

The extrude tool was described in the last chapter, but it can be used in conjunction with other tools to build many different shapes besides simple cookie cutter slabs.

Suppose you wished to build a cylinder with a rounded 45 degree bend--perhaps a model for some plumbing fixture. Before you do anything else, clear the Tri-View with EDIT ERASE ALL. The first construction step is to click on the north window and construct a circle using EDIT ADD CIRCLE. Now move to the West window, expand the Tri-View about three times and select all the vertices. First, use the cursor to indicate the bottom left corner of the West window. Then, execute the menu command TOOLS EXTRUDE. Move the cursor straight across the West window , to the middle of its bottom edge. This will drag out an extrusion of the circle, making a cylinder. Press the right mouse button to disengage the Grabber.

To create the bend in the cylinder, move the Tri-View to the north so that the selected vertices lie near the left side of the West window. Three clicks on the move Tri-View gadget should be about right. Position the cursor directly below the selected vertices at a distance about equal to half the diameter of the cylinder. This point will be the center of curvature of the curved portion of the bend. Execute the TOOLS

EXTRUDE command and immediately click on the clockwise rotation gadget in the edge of the West window. Repeat this sequence of extrusion and rotation eight more times and you will have completed a uniform 45 degree radiused bend. Now all that remains is to extend the pipe at the end of the bend. You could use the Extrude tool again and extend the pipe in a freehand manner.

A more accurate way to extend the pipe would be to use the edge of the Tri-View as a guide, but to do that we must first rotate the whole object by 45 degrees. Make sure the Grabber is turned off and click on the clockwise rotation gadget while holding down the right Amiga key. Only some of the pipe's vertices are selected, so its edges will be tangled horribly. Don't worry, though, we have made an intentional mistake to illustrate something. All is not lost. You just have to rotate the selected vertices back and all will be corrected. To reverse the rotation, click on the counter-clockwise rotation gadget while holding down the right Amiga key. Select the whole object and then you can rotate the entire object clockwise by 45 degrees while holding down the right Amiga key.

Now that all's well, select the end of the pipe again. The selected end should now be lying in the horizontal plane. To complete the object the pipe must be extruded downwards. Using the edge of the Tri-View as a guide, place the cursor on the right side of the West window near the top and execute the command TOOLS EXTRUDE. Drag the cursor down the right hand edge of the window and then turn off the grabber. Finally, call up a lamp, position the observer and target, and create a scanline rendering of your masterpiece!

Creating More Complicated Objects

So far, you have created a number of different objects and seen how even more objects can be created. Still, we have been dealing with relatively simple shapes. There is a reason for this: the majority of complex objects in the real world are actually collections of simple shapes. This may

seem obvious, but this fact deserves special comment because it makes life so much easier. It's better to build a house, for instance, by creating the parts individually and putting them together, than it is to construct the house all in one piece. The key to successfully modelling a complex object is breaking it down into constituent parts that can be easily dealt with.

You might wonder just how to 'put the parts together'. This depends on the purpose for which you will be using the completed object. It would be rather difficult, and in most cases unnecessary, to join individual parts of an object 'physically'--trying to make the objects into a single piece by carefully constructing interconnecting faces and edges. If two objects are close enough together in the scene that you can't see a separation in the rendered image, they are 'connected' well enough.

If you prefer, it's usually all right to let the faces of one part intersect with those of another, as long as it does no harm for a part to protrude through on the other side. In SCANLINE mode SAJr. will sort out intersecting faces for you.

On the other hand, there are some good reasons to tie certain objects together more securely. For instance, you may want to be able to select the whole object with a single SELECT CONNECTED command. To connect parts in this way, just use the EDGE MAKER tool or gadget to create a single edge, tying the two objects together. A single edge between two otherwise unconnected objects cannot make a face, so the new edge will not be visible except in WIRE FRAME mode.

As you design models, you should take care that you choose an appropriate level of complexity. In SAJr., it is possible to model objects to a ridiculous level of detail. Given enough RAM, you could put every scupper on the hull of an aircraft carrier, or place every scale on the wings of a butterfly, but such detail would never be seen in the finished product. To include such great detail wastes time in

rendering as well as construction, and may make the scene too large to render in the available RAM.

As you look at an object you want to model, think of each part in terms of how you could create it. An airplane will have wings, a tail section, a fuselage, etc. Each part may itself be composed of several parts as well--the tail will have a rudder and stabilizer, the fuselage will have a cockpit and engine compartment. At each level of detail, decide whether an item will make any difference in the finished image. You will eventually learn where to draw the line.

As you construct the parts, think of each one in terms of how it resembles EDIT ADD objects, or objects you have built before. Objects will often have several parts that are identical so never create the same part twice if you don't have to. That's what DUPLICATE and REFLECT are used for. By thinking of a scene in terms of its parts, you can dispense with the 'easy' parts quickly, and concentrate on the parts that aren't so easy.

Most of the time, you'll find that the 'hard' parts aren't that hard after all. The primary reason that a new shape looks hard is that you haven't done anything like it before. The best defense against this is to thoroughly familiarize yourself with the tools and commands that we have discussed, and the kind of shapes they can create.

Remember that once you have achieved a basic shape, there are still any number of ways to transform it into the object you want, by adding and deleting vertices, resizing, spinning, etc. Every new object you create will open up a whole new range of creative possibilities.

Notes

NOTES

CHAPTER 7

THE WORLD

This chapter discusses the world in which SAJr. objects reside. Facilities are described for lighting a scene and providing a ground and sky.

Creating Lamps

The subtle nuances of shading that give SAJr. images a sense of reality depend on the illumination of the scene. In Chapter 2, we introduced the EDIT ADD LAMP command which placed a lamp at the current cursor position. The lamp is depicted in the Tri-View as a white star.

There is virtually no limit to the number of lamps that can be created, but since each additional lamp increases the amount of computation required to generate an image, it is wise to try to use as few as possible. It's a happy coincidence that scenes with fewer lamps tend to look better, too.

For dramatic effects, you can employ colored lighting. In order to create a colored lamp, you must first use the EDIT MODIFY LAMPS command. This command calls up a requester similar to the one you used to modify the color of faces. Adjust the sliders to pick the color that you desire. There is no need to dim the light with the value slider. Instead you can enter a number into the Lamp Brightness box to indicate the intensity of your lamp.

The brightness value given to a lamp sets its effect relative to other lamps in the same scene. If you like, you can think of this number as the wattage of the lamp, although the lamp's 'absolute' brightness is irrelevant. As we have seen, SAJr. employs an automatic exposure technique in creating images, and its range of sensitivity is extremely large. For instance, a single ten-'Watt' lamp in a given scene should yield the same image as a single thousand-

'Watt' lamp. If both lamps were in the same scene, however, the ten-Watt lamp wouldn't have much effect. In either case, SAJr. will adjust the exposure so you shouldn't have to worry about the overall brightness of the image.

If you have used the MODIFY LAMPS requester to change the current lamp color or brightness, then any new lamps that are created with the EDIT ADD LAMP command will have the new properties. If you indicate a lamp with the cursor before calling up the requester, you can use the two SET buttons to change either the color or brightness of the indicated lamp. If you wish to reset the properties of all the lamps in the scene, click on the button in the box titled WHICH LAMP and the button will toggle between INDICATED and ALL.

As you did with the colors of the faces, you can use the two sets of FETCH and BLEND buttons to fetch or blend lamp colors and brightness.

In SAJr., as in the real world, the brightness of a lamp decreases or increases according to the inverse square law. This means that if you quadruple your distance from the light, the intensity diminishes to one sixteenth of its original value. SAJr.'s automatic exposure tries to find the best exposure for the most brightly lit object. so As a result, if you place a lamp close to an object, more distant parts of the scene may be very dimly lit. Try to avoid placing lamps much closer to one object than to others,objects unless you want to create some special effect. For an even, overall light, like sunlight, place your lamp five or ten times farther from the target than the observer's location.

If you want to try moving lamp, you may do so with the Grabber tool described in Chapter 4. First, be sure that all vertices in the scene are deselected. Then, indicate the desired lamp and invoke the Grabber as usual.

Try to resist the temptation to add lamps if a scene seems too dark. The automatic exposure feature will compensate for additional lamps, so they will usually not make a scene brighter. Instead, make sure the scene is lit evenly with no

objects too near a lamp. If you still want the scene lighter, use the OBSERVER EXPOSURE MANUAL setting to brighten it.

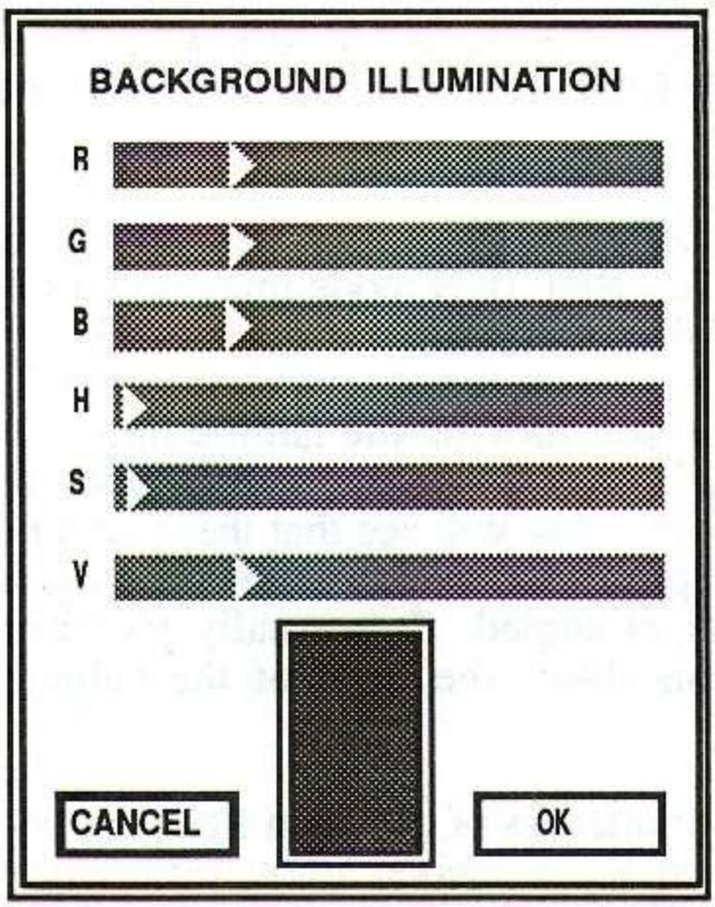
To see the effect of various lighting arrangements, try rendering a few SKETCH mode images of a test subject. A one-subdivision white sphere does nicely. Start with a single lamp at or near the observer's location. Then render a few more images, moving the lamp a little further towards one side each time a picture is taken until the subject is entirely side-lit. You will see that the shading of the sphere becomes deeper and more dramatic as the lighting becomes more oblique, or angled. It is usually good also to raise the main light up above the level of the subject, raising its highlights.

As the obliqueness of the main light source increases, so does the need for a second light--otherwise the shading becomes too deep. The second lamp, called a 'fill' light, will usually be placed to the other side of the observer, perhaps a little closer to the line of sight. The fill light needs to be dimmer than the initial light. While it is all right to reduce its the 'wattage' of the second lamp with the brightness setting, it may be easier to simply place it a little farther away from the subject so you can see at a glance which light has more effect.

These are not rules, by the way, just basic suggestions to get you started. The only rules when it comes to lighting are use your imagination, try things out, and remember the arrangements that please you for use in the future.

Illumination

SAJr. employs a technique called low albedo approximation when it generates an image. This means that any part of an object that is hidden from all light sources will be totally dark. Such objects will not pick up reflected light from nearby illuminated objects. In order to soften the shadows, SAJr. allows you to specify a diffuse ambient illumination that is independent of the lamps that you create.



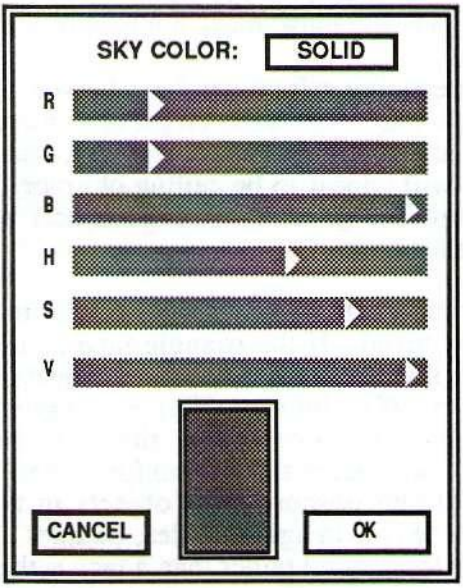
The Illumination Requester

You can change the amount of the illumination with the command WORLD ILLUMINATION. This command brings up an illumination color requester that you operate in the same way as other color requesters. Unlike the lamps, there is no separate brightness facility for controlling the intensity of the ambient illumination, or pervading light, so you must do this entirely with the value (V) slider. Usually the V slider should be set about one quarter of the way from the left, as shown in the above illustration.

The illumination setting can sometimes be used in place of fill lighting, but if the illumination value is set too high, images will seem flat and lifeless. A value that is too small will make shaded faces look unnaturally dark. The default value that is set at program start-up and illustrated above is a good all-around value.

The Sky

You will have noticed a blue background in the images you have created so far. This background, although it is called the sky, need not be blue. Execute the menu command WORLD SKY and you will see a color requester pop up. The SKY requester is similar to the ILLUMINATION requester, but with an extra gadget.



The Sky Requester

Unless it has been changed, the topmost gadget will contain the word SOLID. This a toggle gadget. Click on it a few times and watch it change from SOLID to NONE, then back to SOLID, repeating the cycle as you continue to click.

If you leave this gadget set to NONE, there will be no sky and the background will be black. Notice that the color swatch shows black, despite the settings of the sliders. Select SOLID and see that the swatch now reflects the color

set by the sliders. If the sliders are moved, the swatch changes color normally. If you click OK and make an image, the background color will be the one that you have selected.

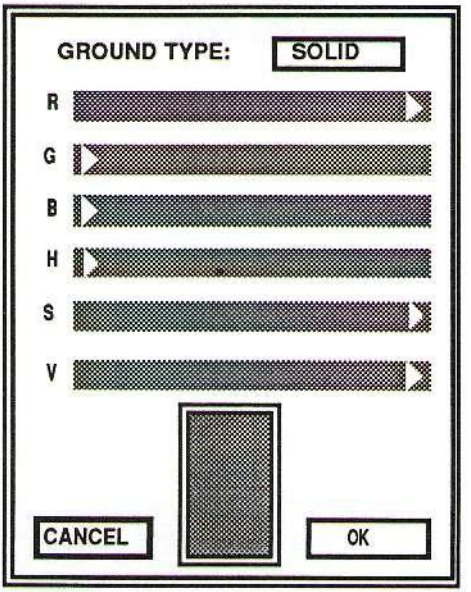
The Ground

If you wish to depict a scene in which objects are not floating freely in space, but are sitting on or suspended near the ground, then you must make a model of the ground.

Some interesting effects can be achieved by adding the ground to a scene. One such effect is to have the ground level above the observer location, target, and the objects. The ground will appear to be ceiling of color. Objects can also pass into the ground making objects appear to be imbedded in the ground.

If the ground is to be flat, then a single triangular face could be employed. If the triangle face is a few hundred times bigger than the scale of the other objects, the face will seem to stretch off to infinity. This sort of ground has some disadvantages: for one thing, there is the bother of constructing and placing the ground for another, this type of ground will often obscure your objects in the SKETCH rendering mode (see imaging modes, Chapter 3). The main reason to use the ground rather than a face is that the ground can stretch to infinity. If you find yourself using a custom focal length lens, you needn't worry that you might see the 'edge of the world'.

As a simpler alternative, you can use the ground plane that is built into SAJr. To use this facility, execute the command WORLD GROUND and a color requester will be invoked.



The Ground Requester

Like the sky color requester, it the GROUND requester has a ground type gadget near the top right corner. By clicking on it this gadget, it can be toggled between NONE and SOLID.

If NONE is selected, then no ground is produced. Otherwise, an infinite ground plane is generated that passes through the Origin. The Origin is the center of the SAJr. universe, and is where the observer's target is initially placed when the program is first started. If the ground exists, and it passes through the Tri-View, it is displayed as a horizontal line in two of the windows. If you enable the ground to your scene but can't see it in a Tri-View window, it is because the Tri-View 'box' is too far above or below ground level. Try expanding the Tri-View until the ground comes into view.

NOTES

CHAPTER 8

CREATING AN ANIMATION

For SAJr. to construct an animation, it must go through an automated sequence of steps. It begins with one or more scenes which you have created to describe the animation's movement and ends with a compressed sequence of images that may be loaded into RAM and played back as an animation.

This chapter introduces a facility called a 'Take'. The Take functions as a control panel which lets you make choices about how the animation will be built.

This chapter will also explain how a few scenes can be used by the Take to create the numerous images that an animation requires. At last you will be able to make your three dimensional objects move.

The Take

In the motion picture industry, a 'take' is a scene or portion of a scene that is photographed without interruption. Clearly a lot of organizational effort is required for a successful take.

In the world of SAJr., the Take is a special file which contains information needed to coordinate and control the construction of an animated sequence. For instance, the Take contains information about where each file associated with the animation is to be stored. The Take also stores data relating to the animation's imaging mode, screen size, etc., so that all of the scene's frames are rendered consistently. A number of other pieces of data are also kept in the take.

Many of the Take's controls may be set directly by you. Other coordinating actions are performed by the Take

automatically as you proceed. These actions and settings will all be explained in this chapter and the next.

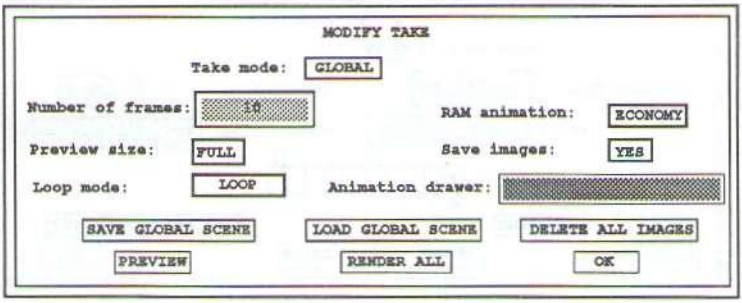
Before you begin working on an animation, you must first specify the name of the Take. This name will be used as a prefix on all of the files that are created in association with this Take, so that different Takes cannot be confused.

To set the Take's name, execute the command PROJECT LOAD TAKE. The familiar file requester will pop up. If you are continuing to work with a previously created Take, then select that Take in the normal way, from the displayed list of files. If you want to create a new Take, just type the new name into the appropriate slot. Click the OK gadget to continue with the new Take. If you change your mind about loading a new Take, click CANCEL to exit to the Tri-View without doing anything.

You will find that you can still use SAJr. just as before. You can load, edit, and save scenes as usual. You can also create and save images. As far as the normal use of the program is concerned, there is no difference.

By naming the take, however, you have activated an extremely powerful command: EDIT MODIFY TAKE, which had previously been ghosted. Execute this command and look at the requester that pops up. The MODIFY TAKE requesters are the control panels for making animations.

When the MODIFY TAKE requester comes up for the first time, it is in GLOBAL mode. At the top center of the requester, there is a toggle gadget which switches the requester between GLOBAL and its other Take mode, FRAME.



Modify Take, GLOBAL Mode

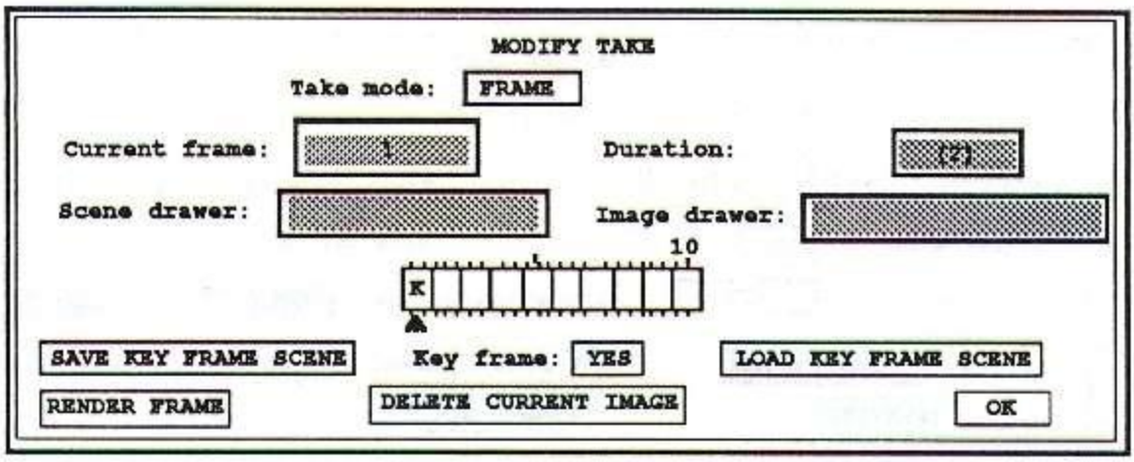
These two modes look radically different, because they permit you to deal with the animation in two entirely different ways. The GLOBAL mode contains settings and commands for the entire animation, while the FRAME mode provides controls that let you deal with each frame of the animation individually, on a frame by frame basis.

To get a better grip on these distinctions, let's examine the requester of each mode. First, toggle the Take Mode gadget to the GLOBAL mode.

It should not be hard for you to imagine that each control that is visible in this mode has some function relating to the whole animation. For instance, the box marked 'Number of frames' is obviously where you type in how many frames you want the program to create for your animation.

Choosing the best number of frames can be a complicated matter, because it involves the desired playing time, display duration of each frame, memory considerations, etc. For the sake of this discussion, type in the number 20.

If you click the Take Mode gadget again, you can take a look at the requester's FRAME mode. In this mode, everything you do affects individual frames of the animation.



Modify Take, FRAME Mode

You should see a row of twenty boxes stretched across the lower middle portion of the requester, looking suspiciously like a strip of photographic film. This is the Film Strip gadget. Each of these boxes represents a frame of the animation. Clicking in a box specifies that frame as the 'current frame'. This will be shown by the number in the 'Current frame' slot, and by a small arrow beneath the Film Strip gadget.

Another control in this mode is the 'Duration' setting. This is the display duration that we mentioned a moment ago. The number in the box is how long, in 'jiffies', the current frame's image will be displayed during playback. One 'jiffy' is 1/60th of a second in the USA and Japan but is 1/50th of a second in most other countries. Longer frame durations make the animation seem slower. We will learn more about this later.

Before we continue, notice that you haven't seen any command to save the Take. All the data associated with the Take is automatically saved whenever you click the OK gadget to get out of the MODIFY TAKE requester. It is also saved when you start an animation rendering.

The Global Scene

In a theatrical presentation the scenery generally remains fixed all the time. The curtain is raised and the actors move around the stage. In SAJr., you can place your scenery,

those elements that do not change, in a scene called the Global Scene.

In the Take requester, the two controls that operate with the Global Scene are the buttons marked SAVE GLOBAL SCENE and LOAD GLOBAL SCENE.

When you click on the SAVE GLOBAL SCENE button, the contents of the Tri-View are written to the disk and become the Global Scene for the Take. This click tells SAJr. to save the scene with a special file name associated with the Take. File names are discussed extensively in Chapter 9.

If you want to work in a previously saved global scene, use the GLOBAL mode button marked LOAD GLOBAL SCENE, then click OK to get out of the Take requester. This command will clear the Tri-View automatically, so you don't need to execute an ERASE ALL before loading the Global Scene.

Key Frame Animation

The animation technique employed by SAJr. is called Key Frame animation.

Key frame animation is very simple and is based upon the observation that there are usually only small differences in adjacent frames of an animation. If the difference is small, you don't have to construct a scene for every frame. Instead, you can supply scenes for certain frames which are known as the key frames. SAJr. automatically constructs the scenes for the non-key frames.

You have to tell SAJr. which frames will be key frames. This is done in the FRAME mode of the MODIFY TAKE requester. You simply select the frame you want by typing a number into the box marked 'Current frame' and then toggling the 'Key Frame' gadget from NO to YES. A letter 'K' will appear in the frame of the film strip image to mark that it is a key frame. If you wish to change the current frame

number, you can also do it by clicking on the appropriate frame of the Film Strip gadget.

Once you have tagged a frame as a key frame, you can click SAVE KEY FRAME SCENE. This saves whatever scene is in the Tri-View as the current frame's scene. A key frame scene is created much like any other, by editing in the Tri-View. However special rules must be followed whenever you create a key frame scene.

We have said that certain frames in your animation will be key frames, so it follows that not every frame needs to be a key frame. The position of key frame objects in non-key frames is derived by comparing the two adjacent key frames, one before and one after the current frame. Specific rules are applied in this comparison and will be explained later.

In Key Frame Animating, there are two methods of motion and change. The first system is much like cel animating where each frame's scene is manually created and may bear little or no resemblance to the the previous frame. This method allows major changes, such as the creation or destruction of an object, to occur instantly. We simulate cell animating by placing key frames directly next to each other on the filmstrip. Manually creating animations this way is a great deal of work as each frame only represents a very short period of time.

The second method of motion and change, Betweening or Tweening, take advantage of SAJr.'s ability to calculate the probable actions of objects based on their counterparts in the surrounding key frames. To do this, the objects in one key frame must bear a strong relationship to the objects in the other key frame. To save you from having to define the relationships amongst the individual vertices, SAJr. remembers their 'Birth Order, or how each vertex is related to its immediate family. The immediate family are those vertices connected directly to the vertex with an edge. If, from one key frame to another, even one vertex severs a relationship (deletes an edge), is born (added), or dies (deleted), the change will affect all of the relationships in the

scene and Betweening will not be possible with those two key frames.

Key frame motion is resolved as SAJr. constructs the 'working scene' for the frame that is being rendered. The 'working scene' is generated by Betweening. If there is a global scene, it will be merged into the 'working scene.'

You can use the LOAD KEY FRAME SCENE gadget on the MODIFY TAKE requester to load in a 'working scene.' This scene represents the state of the animation in the current frame. Pressing the gadget automatically clears the Tri-View, so you don't need to execute an ERASE ALL first. After you load the scene, click OK to get back to the Tri-View.

If there are objects in the global scene, they will now be visible in the Tri-View, but they may not be edited or even selected. If the frame is not a key frame, objects present in the adjacent key frames will also be visible, but cannot be selected or altered.

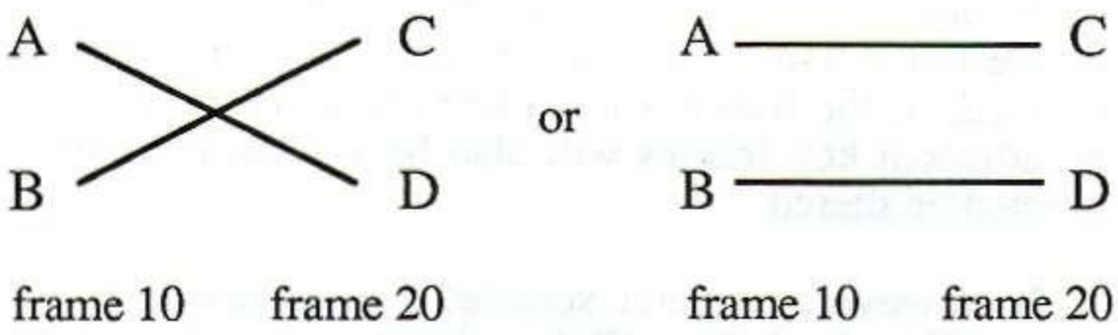
If, however, you have selected a key frame, that frame's scene will be loaded in. Global objects will still be visible so that you can use them as a reference for placing key frame objects. When you have finished editing, call MODIFY TAKE again, and click SAVE KEY FRAME SCENE. Of course, only the key frame objects and not the global objects will be saved.

Betweening

Suppose that you specify frame ten and frame twenty to be key frames and there are no other key frames in that interval. Each key frame has its own scene. When SAJr. must render an image for a frame that is not a key frame, frame thirteen, for example, it interpolates between the key frames that bracket, or lie to either side of, the frame. In this case, frame ten and frame twenty are our bracketing key frames.

Suppose that each key frame contained only a single vertex. The interpolation process, or Betweening, would assume that the vertex moved smoothly from its position in frame ten to its position in frame twenty. In frame thirteen, the vertex is placed 30% of the way along the path.

So far, so good. Now let us complicate the scenes and place two vertices in each key frame. The Betweening process can be carried out as before, provided that we know which vertex in frame ten corresponds to which vertex in frame twenty. It is up to you to supply the clues that let SAJr. resolve the dilemma. Suppose the two vertices in frame ten are called A and B, and the two vertices in frame twenty are called C and D, then which of the following do you want?



The Betweening Dilemma

The second might seem preferable, in that the paths do not intersect, but in three dimensions, the majority of lines do not intersect. If we had more than two vertices, it would become still harder to decide which vertex in one frame corresponded to a vertex in the other frame.

You could be required to assign a unique name for each vertex, but that would be a painful chore if your scene had a thousand or more vertices. Fortunately, this is not necessary.

It turns out that the vertices and other parts of a scene are stored in a particular order as they are created. We will call this order of creation the Birth Order. This order remains the same in the scene as long as nothing is added or deleted.

More significantly, the order remains the same while vertices are moved.

In order for key frame animation to work, you need only make sure that consecutive key frame scenes are modified copies of the same scene.

Although this may sound complicated, it could not be easier in practice. Suppose that our animation has one key frame every five frames, so that the key frames are 1, 6, 11, 16 etc. In the MODIFY TAKE requester, go through and mark every fifth frame as a key frame. Now use LOAD KEY FRAME SCENE to load frame number one. There is no key frame data yet, so only the global portion of the frame will be visible in the Tri-View. Create the scene using any of the regular SAJr. tools, and save the scene with SAVE KEY FRAME SCENE when you are finished.

Scene number one has been saved now, but a copy of it is still in the Tri-View. You can manipulate this scene, move or rotate objects, do anything that you wish, but do not create or erase any lamps, vertices, edges or faces.

If you somehow alter the scene's structure, say by accidentally adding or deleting a vertex, just use LOAD KEY FRAME SCENE to load in the first frame again.

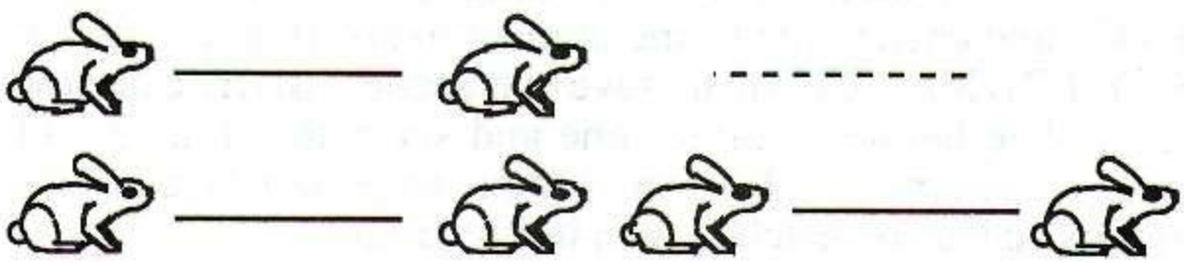
When the scene is to your liking, go back to MODIFY TAKE and change the frame number to six, then use SAVE KEY FRAME SCENE to save the scene. SAJr. can now interpolate between frames one and six with no danger of ambiguity, since each lamp, vertex, edge and face in key frame one has its matched twin in frame six.

You can apply this technique to bridge the gap in any two key frames that are separated by a number of non-key frames. There is a quick way to check a pair of key frames to make sure their structures match. Just select any non-frame bracketed by the key frames to be compared and call LOAD KEY FRAME SCENE. Since this is not a Key Frame, SAJr. won't load the frame. Instead it will say: **Not a keyframe! Generate scene betweening?** If you

respond OK, SAJr. will compare the two Key Frame's relationships and move the objects the appropriate distance. If the structure of the scenes is different, you will see the message: **Keyframes 1 and 6 have different structure.** This means that the two key frames differ in a way that precludes the possibility of Betweening. If there is only one key frame in the segment when LOAD KEY FRAME SCENE is called, SAJr. will merely load a duplicate of the key frame into the Tri-View.

By now you may think that all of a Take's key frame scenes must have the same structure, the same number of vertices, etc. Fortunately, this is not always the case. This strict rule permits Betweening, but if two adjacent frames are both key frames, no Betweening takes place. Two key frames placed together may be as different as you like.

Suppose that you have two animated rabbits, and you want one of the to magically vanish after frame 20. Build the scenes as usual for the key frames before frame number 20. Make sure that frame number 20 is a key frame, and save its scene. Now make frame number 21 a key frame also. Erase one of the rabbits in this frame's scene and save it as key frame 21. Continue to construct the subsequent key frames, ensuring that their structures match the scene from frame number 21.



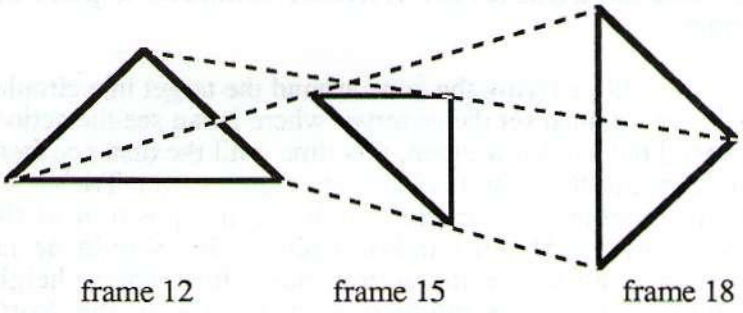
The Vanishing Rabbit Trick

The rules of Key Framing are simple. If you want SAJr. to Between frames, don't add or delete vertices, edges, or lamps in your bracketing key frames. The vertices or lamps

may be moved, however. In the event that additions or removals are required, you must use two adjacent key frames to bridge the transition.

Betweening Distortion

The Betweening process is very simple, involving linear interpolation, but it can produce strange effects. Consider the following case: We have two key frames, frames number 12 and 18, with no key frames separating them. A triangle in frame 18 is identical to the corresponding triangle in frame 12, except that it has been rotated 90 degrees. The following illustration shows the Betweening process for frame 15.



Betweening Distortion

You can see that at frame 15, the triangle generated by the Betweening process is much smaller then the two identically shaped triangles in the key frames. This is caused by the each of the vertices trying to travel in a straight line approximation of the 90 degree arc. Rotating an object 180 degrees in only two key frames may cause an object to literally turn inside out! This Betweening distortion may be minimized by inserting more key frames. The additional key frames allow you simulate the arcs and curves more precisely and ensure that objects do not change too much from one key frame to the next.

An Example Of Animation

Let us make an animation of an orbiting cube using key frame animation. First, clear the Tri-View with EDIT ERASE ALL. If you have been working with a Take, you will be asked if you wish to erase it. Answer OK to this--nothing will be lost, the Take still exists safely on disk, you are just erasing the copy that is held in memory.

Next, add the cube to the Tri-View and click the Expand Tri-View gadget until the cube is about one eighth of the size of the Tri-View or about four clicks. Indicate the position just above the Move Tri-View gadget in the middle of the Down window's south edge with the cursor.

This point will be the animation's center of interest. Execute the OBSERVER TARGET command to place the target.

We will be flying the cube around the target in a circular path, so we must set the observer where it can see the action. Expand the Tri-View again, this time until the distance from the target to the cube is about one eighth of the Tri-View's width, another four clicks. Indicate the position at the bottom of the Down window again. This should be far enough away to see the action, but a little added height would give a more interesting view. Go to the North window and move the cursor up by about one third from the top edge of the Tri-View. Execute the command OBSERVER LOCATION.

Place the cursor on the target--this will require moving it in two windows. Next click on the Center Tri-View gadget, then click Contract Tri-View gadget until the cube is just about to leave the Tri-View.

When you 'zoom in' with Contract Tri-View, the cursor and target might seem to drift apart. This is because contracting the Tri-View increases its resolution, making a small error in placement seem larger. If this happens, just move the cursor back to the target again.

Now we are ready to set the cube in motion. Select the cube with the EDIT SELECT ALL command. Execute PROJECT LOAD TAKE to name a new take, then call up the MODIFY TAKE requester and enter 36 as the number of frames in the global mode. Remember that you need to click in the box, change the number, and press the return key to enter the new value.

Since our cube is to fly around a full circle (360 degrees) in 36 frames, it must move 10 degrees per frame. Let us make every fourth frame a key frame. Click on the Take mode gadget so that it becomes FRAME. Click on the Film Strip gadget for frame number 1 and click on the Key frame gadget so that it changes from NO to YES. A letter 'K' will appear in the first frame of the film strip image to mark that this is a key frame. Now click on the fifth frame and again click on the Key frame box. Repeat the process until frames 1, 5, 9, 13, 17, 21, 25, 29, 33 and 36 are all key frames.

It doesn't take a mathematician to see that frame 36 is only the third frame past 33, but there's a good reason for this inconsistency. The circular path has a 10 degree arc linking the frames. Of course, one of these arcs is the one linking frame 36 and frame 1. This arc had its effect when the animation looped, making a smooth transition from one cycle to the next. Our key frames don't specifically account for this last 10 degrees, but if we just leave off 10 degrees 'short' at frame 36, everything will work when the animation loops. Honest! If we had actually linked the frames with this final arc, frames 1 and 36 would be identical and our animation would seem to pause for a moment.

Examples are for doing, so let's create the scenes. Click on the first frame in the film strip gadget, then click on the SAVE KEY FRAME SCENE button. This saves the scene in the Tri-View as the first key frame. Now we must construct the next key frame scene, which is frame 5. Click on the OK gadget in order to leave the requester.

Back in the Tri-View, make sure that the cursor is lined up over the target, and that the cube is visible near the top of

the down window. We need to rotate the cube around the target in a clockwise direction by 40 degrees. You can either click on the clockwise rotation gadget in the Down window 8 times, or click on it once with the right Amiga key held down (for a 45 degree rotation) and then click the counterclockwise gadget in the Down window once (to back up by five degrees).

Return to the Modify Take requester with an EDIT MODIFY TAKE command. Click on the Film Strip gadget at frame number 5 and press the button SAVE KEY FRAME SCENE.

Repeat this process for each frame, rotating the cube around the target and saving the key frame scene. Remember we need to leave frame 36 short by 10 degrees, so only rotate the cube 30 degrees (6 clicks) for this final frame.

Once you have saved all the key frames, all you need to do to see your handiwork is toggle to the GLOBAL mode and click PREVIEW.

Previewing An Animation

Later in this chapter, we'll see how to render an animation in full color, and compress the images for playback from RAM. The compressing process can be time consuming, especially for large animations. In this particular case it would be unrewarding as well, since we've neglected to put a lamp in the scene.

Right now, however, all we need to see is the motion of the scene, and we aren't particularly interested in keeping the results for posterity. For this purpose, SAJr. provides quick preview facility. This facility uses WIRE FRAME mode in low resolution, so it can assemble the animation in a minimum of time and memory.

To start the preview, execute the EDIT MODIFY TAKE command and make sure that the Take mode is set to

GLOBAL. Then click the PREVIEW gadget. Each frame of the animation will be displayed as it is calculated. When all the frames are done, the entire animation will be played repeatedly.

The initial speed (frame rate) of the playback will depend on the duration that you have set for each frame of the animation. In our example, we haven't changed the default setting of 2 jiffies per frame, this means each image will be shown on the screen for 1/30th of a second (1/25th of a second on PAL Amigas).

While the preview is being played, you can press the keyboard's top-row digit keys 0 through 9 to modify the speed as follows:

Key	Speed
0	normal rate
2	2 jiffies per frame
3	4 jiffies per frame
4	8 jiffies per frame
5	16 jiffies per frame
6	30 jiffies per frame
7	60 jiffies per frame
8	120 jiffies per frame
9	240 jiffies per frame

Pressing the spacebar will momentarily halt the playback, pressing the spacebar again resumes it.

When you want to stop the preview entirely, press the right mouse button or the ESC key. The preview screen will be closed and you will be asked if you want to unload the preview.

If you think you will want to view the preview again later in the session, you can answer NO to this question. This causes the preview to be kept in memory. It can be replayed from the Tri-View by executing the command PROJECT SHOW PREVIEW.

The disadvantage of retaining the preview data is memory is that you will have less memory for other things. You can free up this memory at any time by executing the command PROJECT UNLOAD PREVIEW.

If your animation is very long, the preview display may fail because of insufficient memory. In that case you can return to the MODIFY TAKE requester, select GLOBAL mode, change the 'preview size' gadget from FULL to MEDIUM, and try the preview again. Because the image generated has one fourth the area, the preview will only take about one fourth the memory.

Never hesitate to use the preview facility, it is much faster than generating a full size, color animation, and it will often show you ways to improve your object motion and placement, as well as your observer's position. The time you save will be well worth this extra step.

Looking At Individual Frames

If you want to see what this 'working version' of the scene looks like in a particular frame, call the MODIFY TAKE requester and click to the FRAME mode. Select any frame of interest, click LOAD KEY FRAME SCENE. This command automatically clears the Tri-View, so you don't need to execute an ERASE ALL first. Now click OK to get back to the Tri-View.

What you will see is the scene that SAJr. will use to create this frame's image. You can now move and resize the Tri-View to examine various parts of the scene. This allows you to make sure that the animation is proceeding precisely as you expected.

When you are looking at an individual frame's working scene, you will find that some Tri-View commands will not work as you may expect. For instance, none of the selection or editing functions will work on the global scene's objects. If you want to render an image of a frame's working scene, select that frame in the FRAME mode of MODIFY TAKE

and then click RENDER FRAME. The image will be rendered in the size and imaging mode that currently applies for the take. This will be what was set for the last global or key frame scene that was saved before the last DELETE ALL IMAGES command.

If you want RENDER FRAME to match the preview images, set the imaging mode to LO-RES, NO-INTERLACE, WIRE FRAME.

RAM Animation

So far, all of the animations you have done were WIRE FRAME previews. These two-color images are small enough in memory that they can be stored directly in RAM and 'page-flipped' to the screen for animation. We've seen that SAJr. can create much more beautiful images in full color and much higher resolution. The more impressive images have a cost, though, in the amount of data they contain.

An image rendered in HI-RES, INTERLACE, PAINTING, for instance, can take over 16 times the memory that a preview image does. Typically, color images occupy from 50,000 to 100,000 bytes. To show a single second of animation, then, involves moving about 2 MegaBytes of data into the Amiga's 'chip' graphics RAM every second. This data rate is more than the Amiga's processor can handle, even if the whole animation could be loaded into expansion memory and copied from there.

Fortunately, in most cases, not all of the image data needs to be moved. One frame in an animation is usually very similar to its predecessor, so only the differences of the frames need to be changed. This technique was used to produce the Juggler demonstration in 1986, and has since become known as 'delta' compression. (The Greek letter delta is often used to symbolize change or displacement.) The data describing the changes from one frame to another is called 'delta data'.

In delta compression, an animation is reduced to an initial image plus a series of data chunks that can be used to reproduce each subsequent image by altering the current one. A standard file format, called the ANIM format, has been suggested for animation files. SAJr. employs a variation of this format called J-type animation. At this writing, at least three other ANIM formats are in use.

RAM animation is not without its drawbacks. Because two consecutive images are compared on a pixel by pixel basis, some animations may not be amenable to delta compression. The problem occurs if a very large part of the image changes from one frame to the next, for example when large objects pass very close to the camera.

Even more extreme cases are when the observer's location or target are moved, when the observer lens's focal length is changed, or when lamps are moved. These are advanced techniques that will be discussed in Chapter 9, 'Advanced Animation Techniques'. Each of these cases virtually guarantees that nearly every pixel of the image will change. Since some overhead is involved in compression, this can lead to 'compressed' anim files that are larger than the original image data.

Too much motion will not only cause extremely large anim files. It increases the amount of data that must be moved by the processor which will slow the playback. In a typical application, if you do not achieve a compression of at least a three-to-one ratio of the image size to the delta data size, the animation is unlikely to replay at the desirable rate of two jiffies per frame (25 or 30 frames per second).

The other obvious restriction for RAM animations is that the amount of memory on your machine limits the total duration of the animation. If each image is moderately different from the next, you can expect to play about three seconds of animation for each MegaByte of available memory. If only small portions of an image are changing, the duration could be as long as 30 seconds per MegaByte.

The apparent length of an animation can be increased by designing in such a way that the animation repeats itself in a short length of time. Such looping animations are obviously less interesting than a full length feature film, nevertheless, impressive animation demos have been made in this way.

You need not restrict yourself to a single object moving in a circle. For example, you could produce a scene of a column of troops marching along a twisting road. In the time that it takes one figure to march the distance from himself to the person in front of him, the scene can be made to exactly repeat itself. An animation can also be extended by carefully placing pauses in the action. The pauses are in the form of particular frames with long duration values. Although RAM animation does impose some severe restrictions, there is no limit to the creativity that can be applied to this new art form.

RAM animations serve two purposes. As a stand alone end product, they may be acceptable for short animations, a spinning logo for a commercial, for example. Alternatively, short animation Takes may be assembled into longer sequences by combining them on video tape using standard videotape editing techniques. This probably represents the most cost-effective approach to computer generated video animations.

Making A RAM Animation

The button marked RENDER ALL in the GLOBAL mode of the MODIFY TAKE requester starts the animation images rendering. The preparation required for this process is essentially identical to that for a preview, but there are a few additional items that need to be attended to.

First, you obviously need to consider color, and lighting in preparing your animation. The examples created so far didn't even have any lamps, you should place at least one in the global scene. Everything you have learned about creating good images applies here. Don't fail to try a few

tests before committing to the full render. Look for good color, lighting, and composition.

You will also need to remember to set the imaging mode and size for your animation. These settings are taken automatically from the last scene saved with either SAVE KEY FRAME SCENE or SAVE GLOBAL SCENE.

An animation may consist of dozens or even hundreds of images, and take days to render. Because of this, be sure to render a few images here and there through the animation, using the RENDER FRAME button in the FRAME mode of MODIFY TAKE. This command will render the frame's image in the current imaging mode, as discussed before, and then save the image to disk.

Finally, it is a good idea to click DELETE ALL IMAGES before starting a full render. If you have used RENDER FRAME in setting up your animation, then some undesirable images may remain on disk. These will not be re-rendered. Also, the color map and imaging mode may have been 'locked'. The meaning of this will be explained in Chapter 9, but for now just remember that DELETE ALL IMAGES resets the Take and makes it ready to start a new render.

If you are sure that the color map and imaging mode are OK then you can skip the DELETE ALL IMAGES. This avoids having to render those images twice. If you want to delete one image, but not all of them, use DELETE CURRENT IMAGE in the FRAME mode.

When you finally click RENDER ALL, a process will begin that closely resembles the PREVIEW render. We will look at this process as it would occur normally, with the GLOBAL mode controls (except number of frames) set to their defaults. We'll discuss variations also in Chapter 9.

First, SAJr. will go through the animation frame by frame, rendering all the images. These images will be saved out to disk individually, by default in the same directory that was specified in the LOAD TAKE requester. Unless the animation is fairly small in size, you will need a hard disk to

handle all this data en masse. Alternatives for advanced floppy disk users are discussed in the next chapter.

Once all the images have been rendered, SAJr. will again proceed through the animation. This time it will go through leapfrog fashion, finding the differences in alternating pairs of images. It starts as you might expect, with frames 1 and 2, but then it goes to 1&3, 2&4, etc. The images are compressed this way to facilitate 'double-buffered' playback. Double buffering keeps two images in memory for the best possible speed, one is updated while the other is displayed.

As each delta is found, it is written out to the animation load file, or 'anim' file. This file will contain the complete compressed animation, ready to be loaded and played back. The anim file is also saved to the Take directory by default. Again, consult the next chapter if this is not desired.

The compression phase can be quite lengthy, since the REGULAR compression scheme was designed for greatest compression and fastest playback speed, not shortest compression time. You can use a much faster compression scheme by clicking the MODIFY TAKE requester's RAM animation gadget from YES to ECONOMY, its second state. ECONOMY compression is much faster in calculation, but generally creates larger anim files that may not play as fast. Both these compression schemes employ the so called 'J-op-codes' in the ANIM file. An alternative is to use the third option that is provided by clicking on the RAM animation gadget. The gadget changes to ANIM-5. The animation file will be written using 'op-code-5'. Although this method of compression often yields slower playback, the compression phase is fast, and this animation format is supported by several third party software products. When ANIM-5 is selected, all animations will loop continuously, the oscillation facility is not available. Some animation player programs are unable to show animations produced in the MEDIUM or OVERSCAN image sizes.

When the animation render is finished, you can click OK to get back to the Tri-View. If you have to interrupt the

render (by pressing CTRL-C), or if it is stopped for any other reason, any images that have been saved will still be present on the disk, and will not need to be re-rendered. When you click RENDER ALL again, the Take will find the images and rendering will pick up where it left off. If the compression phase is interrupted, compression must start again at the beginning.

Playing A RAM Animation

When an animation is constructed, an icon (the file with the '.info' extension) will be created so that the animation can be played by double clicking on its icon. The program that actually plays the animation is called **movie**, and it must be present in order to play the animation. Because the animation player program must take control of the Amiga hardware in order to display a moving image, **no other task must access the screen while the animation is playing**. If you violate this rule, memory may be corrupted with unpredictable results. You could even lose data on your disks!

The easiest way to ensure that this problem never occurs is to avoid the Amiga's multitasking capabilities. Unless you have a lot of memory, you will want to shut down all your other tasks in order to make as much memory available as possible for the purpose of displaying the animation. Beware of hidden tasks that can be invoked with a keystroke, or those public domain 'display hacks' that make the cursor behave in a funny way.

Under no circumstances attempt to play one animation while SAJr. is generating another animation.

The animation player program, **movie**, is freely distributable, so if you give a copy of your animation to a friend, remember to include a copy of the player program also.

Notes

NOTES

CHAPTER 9

ADVANCED ANIMATION TECHNIQUES

The last chapter walked you through the basics of animating scenes with SAJr. The discussion has focused on the simplest conditions, leaving many of the available options in their 'default' state. This chapter describes these additional options, which you can use as you gain confidence in creating animations.

Also in this chapter are sections which introduce some advanced animation techniques.

Loop Mode

One of the options we have ignored is provided by the MODIFY TAKE gadget marked 'Loop mode:'. This option applies to both RAM animation and to PREVIEWS.

The default setting for this gadget is LOOP. We have already seen this mode in action. In it, the animation starts over again after the last frame is displayed. We have also seen that the last and first frames should usually not be identical, otherwise the motion will seem to pause for an extra frame when the animation loops. A five frame animation would loop as 1, 2, 3, 4, 5, 1, 2, 3, etc. When LOOP is in effect, RAM compression will not stop at the last frame, since deltas must be calculated to carry the animation back through to frames 1 and 2.

An alternative way of producing an animation is to have it replay in the form of an oscillation. Set the loop mode gadget to OSCILLATE before rendering the animation. A five frame oscillating animation will play the frames in the following order: 1, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3 etc.

If the loop mode gadget is set to NONE, then the animation will simply play to the last frame and stop.

Frame Duration

We have already mentioned frame duration, but it is important enough that it deserves a fuller explanation.

The 'duration' of a frame is the time, in jiffies, that the frame's image will be displayed during playback. The duration might actually be longer than what you set, since some animations are too complex to play back at full speed.

The definition of 'jiffy' can vary depending on where you live. In the United States and Japan the video standard is NTSC. NTSC displays each image as two interleaved fields, at 60 fields per second, so an NTSC jiffy is 1/60th of a second. In Europe, Australia, and elsewhere, the video standard is PAL. A PAL video jiffy is 1/50th of a second.

When you first invoke the FRAME mode of the MODIFY TAKE requester, you will see that the duration of frame one is shown as (2). The parentheses indicate that the value is derived rather than specified. If you set a particular value, say 4, then the value is shown without parentheses. Now look at the second frame and it will be shown as (4). Unless you explicitly set a duration value, a value will be derived from the nearest previous frame that has been given a specified value. If a frame's duration is set to zero, then the frame will assume the value derived from an earlier frame.

This behavior allows you to set the duration for a large number of frames at once. If you click in the duration box and delete the old value (be sure to delete parentheses, too), then you can type in a new number. This number will effect this frame and all subsequent frames that have not been otherwise set.

One note of caution: If you want to set a unique duration for a frame located somewhere in the middle of the

animation, you will need to set the next frame's duration back to the default. Otherwise, the rest of the animation would take on the new value.

Frame duration is directly linked to both the apparent smoothness and the playing time of the animation. In NTSC, a 30 frame animation will last only one second at two jiffies per frame. If you make the duration longer in jiffies, the animation will play longer, but will not be as smooth.

File Names Conventions

If you use the Amiga's command line interface (CLI), you will probably have noticed that SAJr. creates image files with names that end in '.image', and scene files that end in '.scene'. This is how the program is able to tell different types of files apart.

You will not be surprised to learn that a Take file ends in '.take'. The Take file is quite small because it does not actually contain any scenes or data. Instead, the Take file accesses separate scene and data files.

When you name a Take, you are also indirectly naming all of its scenes. If the name of the Take is 'xyz', its file name is actually 'xyz.take'. The take's global scene file is called 'xyz.g.scene', and the first key frame scene is called 'xyz.1.scene'. If frame 20 is a key frame, its scene would be called 'xyz.20.scene'. When images are generated, they are named in a manner similar to the Take and scene. The name for the image for frame number 20 is 'xyz.20.image'.

This simple naming convention is carried out automatically, so you don't have to worry about it. However, when you are using the Amiga's CLI to maintain your directories, it makes it easier to distinguish between the files of various Takes. The Amiga's LIST and DELETE commands permit the use of patterns. For instance, the command

LIST p xyz.#?

will list every file associated with the Take named 'xyz'.

Note that SAJr. does not save '.info' files for the files it creates, except for animations, so the files will have no icons on the Workbench screen.

SAJr. can easily create hundreds of files on a single directory, so icons would quickly become useless. The '.info' files themselves would effectively double the number of files, drastically slowing disk access for some operations.

If you haven't done so already, you will find it valuable to study your Amiga's CLI documentation before going very far with SAJr. The definitive source for this information is the The AmigaDOS Manual, from Bantam Books, which is available through most bookstores where computer books are sold. This book fully explains the Amiga's file system and CLI. There are also several novice-oriented AmigaDOS texts on the market. Be certain that the AmigaDOS text you choose describes the most current version of AmigaDOS.

Disk Space

In the course of generating an animation you may find yourself dealing with many MegaBytes of data, so a little advance planning is wise. At times, you may need to estimate how much disk space each image will take.

SAJr.'s images are stored in a compressed format, which can vary in size from image to image, so you will need to check the size of an image from your particular animation. To do this, use RENDER FRAME to generate a test image before you click RENDER ALL. Try to pick the 'busiest' frame--an image with many transitions from one color to another will generally use more disk space, and be a 'safer' indicator of what's needed.

Once RENDER FRAME has saved the image, go back to the Workbench screen and the Amiga's CLI window. Use

the CD command to change your current directory to the drawer where the image is stored (the same drawer where you opened your Take). Then use the AmigaDOS LIST command to see how many bytes are in the test image.

In the Amiga's operating system, disk storage is measured in 'blocks' of 512 bytes. Typically, because of system overhead, only 488 bytes will be useable per block. A newly formatted Amiga floppy disk can hold 1751 blocks of data (1758 if the 'noicons' option is used).

To see how many blocks are used by your test image, divide its number of bytes by 488, round up, and add 1. To see how many images will fit on a floppy, divide the number of blocks into 1751, round down, then subtract one to be safe.

If the floppy is partially full, or you will be using a directory on a hard disk or some other device, the CLI "INFO" command will show how many free blocks remain on each device in the system.

Organizing Floppy Disks

So far, our discussion applies to using SAJr. on hard disk systems as well as floppies. While the hard disk is preferable by far, it is possible to render and compress RAM animations using only floppy disks.

The key to this endeavor is the old maxim: a place for each thing, and each thing in its place. You will need to find room for the Take and its associated scenes and images, as well as a place for the final animation.

SAJr. makes use of 'overlays', segments of code that are loaded in at various times, depending on what is being done. This reduces the amount of memory required to run the program, but it means that the program will need to consult its own disk at times during a session. SAJr. also uses code libraries kept on the Workbench disk. To reduce the amount

of swapping required, you can 'boot' from your working copy of the SAJr. disk (insert it instead of Workbench).

The Take and scenes will usually fit on the floppy disk with SAJr., so they're no problem. If the scene files need to be placed on separate disks, the procedure will be much the same as for image files, which we will discuss in detail.

Image files are definitely a problem since as few as five files might fill up a disk. A full disk error will bring the rendering process to a halt. The animation file is also a problem, but we'll discuss that in a moment.

To avoid running out of space while rendering, use the formulae from last section to figure out how many of your animation's images will fit on a disk. Before you start the animation, format (initialize) enough disks to hold all of the frames' images. Name each of these disks with a short, unique name, like 'images1-5', 'images6-11', etc.

Now, how is SAJr. to know which image goes on which disk? You are able to specify a disk name in the 'image drawer' gadget of MODIFY TAKE, in the FRAME mode. Select each frame in turn and type in a disk name for every frame. Use the 'frame drawer' gadget if you need to do this for key frame scenes.

Do not type 'df0:', or any other device names, into these drawer gadgets. This does not give SAJr. any way to ask for the correct disk. Use only the unique volume names of one of the floppies you have formatted. Remember, however, that a disk's volume name must be entered into the gadget with a trailing colon, e.g., 'images1-5:'.

The image and scene drawer gadgets work similarly to the duration gadget, in that any name you enter will be carried over to any subsequent frames without a specified image drawer. This reduces the amount of typing that has to be done. A derived name (one that is carried over) will be shown in parentheses. Be sure that you erase the parentheses whenever you enter a new value.

As SAJr. renders the frames it will ask for the next disk by name when the time comes to write an image to that disk.

When all the frames have been rendered, SAJr. will begin to compress the image data. This is when the anim file must be opened. It is best to have a blank floppy disk dedicated to the animation file, since it can be very large.

This disk should be formatted beforehand, and its name entered in the 'animation drawer' gadget of MODIFY TAKE, GLOBAL mode.

As the animation file is constructed, SAJr. will again ask to see each of the image disks in turn. If you have only one floppy drive, this will undoubtedly incur a painful session of disk swapping since the anim disk will also be needed for each frame. If you have two floppy drives, you can generally place the anim disk in one drive, then cycle the image disks through the other. This avoids much, but not all, of the disk swapping.

There is a practical limit to the size of an anim file: about 850K. This is the maximum size of any file on an Amiga floppy disk. It is possible that larger animation files may be assembled on hard disks, but they must be split into portions to be transported on floppies. Public domain split programs exist which will perform this function. The AmigaDOS JOIN function can be used to re-assemble a split anim file.

Saving Images

You can avoid a great deal of the trouble of working with floppies if you set the 'save images' gadget in the MODIFY TAKE requester to NO.

As we have seen, each image of an animation is normally saved to disk when it is finished. With save images set to NO, the animation will be compressed as it is rendered. Images will still be written to disk, but the image files will be deleted as soon as they are not needed. At most, five images will be retained at one time.

The drawback to this kind of animation is that any interruption will make it necessary to re-start the entire process from the beginning.

Color And Exposure Locking

There are two variables in a scene that can cause radical changes in the appearance of an image. These are the scene's exposure and the image's color map. Both of these items are determined automatically when you render an image with START, and are usually no cause for concern.

If each image were computed with an independent automatic exposure value and one object were to move closer to a lamp, then the overall brightness of the scene would be reduced to prevent 'burn out' of the object near the lamp. This would probably be a desirable action, except for its effect on delta compression. If the overall image is darkened, most pixels would differ from one image to the next and the efficiency of delta compression would suffer.

When SAJr. generates an image, the image is restricted to a color map of 16 or 32 colors, depending upon the resolution used. Each color is chosen from a palette of 4096 possible colors in such a way as to best represent a particular scene. The process of delta compression requires that each frame uses the same set of 16 or 32 colors.

When the first frame of an animation is rendered in SKETCH mode, SAJr. will determine the best color map and exposure for that image. These values will then be 'locked' throughout the animation.

You can have a little control over the color map that will be chosen for your animation. If RENDER FRAME is used to render the most representative frame of the animation, that frame will determine the color map and exposure. If you change your mind and want to start from scratch, click the DELETE ALL IMAGES button. This resets the locks so that the next frame rendered will determine these values.

When an image is rendered in SCANLINE mode, each image is analyzed to determine what is the best color map. Finally, the images are constructed using this optimum color map.

If, for some reason, you want the exposure value to change during the course of the animation, you can 'animate' the observer's manual exposure setting, saving different values for different key frames in the animation. This will usually make the compressed RAM animation much larger. The manual exposure control can also be used to change the overall brightness of the animation.

Animating The Observer

Dramatic effects can be achieved by moving the observer's location and target during an animation. The visual effects produced this way can be very exciting. Often, more information can be conveyed in this way than by using a static viewpoint.

In key frame animation, the location and target may simply be placed differently in different key frame scenes. The location and target will move smoothly from one position to the other when animated.

Other observer attributes can be animated in key frames, as well. Specifically, the lens and the exposure may be animated. Simply set these parameters differently in the key frame scenes, and they will change smoothly in the animation. As usual, key frame animation gives you the maximum creative freedom, at the cost of a little extra work.

The only drawback to animating the observer is that the compression ratio for RAM animation is reduced, often dramatically, because most of the pixels that make up each image will change from frame to frame. This may be of less concern if you have a hard disk or other enormous storage medium.

When such drastic differences exist between frames, the benefit of normal RAM animation compression is reduced. In this case, you might as well click the RAM animation gadget to ECONOMY to speed things up.

When the compression efficiency is reduced, of course, the animation playback speed will suffer as well since the processor will have that much more data to move between frames.

Animating Lamps

The effect of moving lamps in a scene can be equally dramatic, although the cost in processing and compression can be even higher than that of moving the observer. To animate a lamp, place it in the key frame scenes in the desired positions. You can also change the brightness and even the color of each lamp. If multiple lamps are used, in-betweening problems can occur unless you drag the lamps with the Grabber rather than creating new ones for each scene.

Model Animation

In the movies, a classic animation technique is to animate models. A single picture is taken and then the model is moved a small amount and a second picture is taken. The process is repeated until a complete animation has been assembled.

This technique can employ many different types of models. Frequently the models are rigid, but with many jointed parts that can be set in particular positions. Alternatively, a malleable medium like clay can be used, which may require more skill but has almost unlimited creative potential.

In SAJr., solid models may be built with the parts that will be moved unconnected one from another. The SELECT CONNECTED command can pick out one or more parts to be manipulated with the Grabber or rotation gadgets.

A complete human model could be built from many parts, and partially selected. For example, the left forearm and hand could be selected. By placing the cursor at the elbow, the rotate gadget could pivot the lower arm. Then the entire arm could be selected and the cursor moved to the shoulder for more rotation. With a little skill, the human figure could be made to salute. The labor of animation is reduced by employing key frames. A walking man animation requires fairly frequent key frames, say one every four or five frames, but that is still much better than having to build every scene.

It is a good idea to place a sequence of vertices connected by edges in the global scene. These curves act as guides for positioning the model and help to ensure that the motion is smooth and fluid. Just as the eye can detect irregularities in a hand drawn line, it can also detect irregularities in motion. Since the guide curves have no faces, they will not show up in images generated in the SKETCH or SCANLINE modes.

Objects should not normally change their velocity instantaneously. Have velocity build up and decrease steadily. Remember that velocity is determined by the difference in position of a vertex from one frame to the next. The greater the difference, the greater the apparent velocity.

Conclusion

Now that you have mastered SAJr. you can use the reference manual in the appendix if you forget how a particular command works.

The scenes that you can build with SAJr. are limited only by your imagination and the amount of memory on your Amiga. Enjoy the product and enrich the world with your creations!

NOTES

APPENDIX A

SAJr. FUNCTIONAL REFERENCE

The Geometry Of SAJr.

The Tri-View is a rectangular block of space that is represented by three windows. These windows show wire frame projections of the objects contained within the block of space, each showing a different point of view. The size and proportions of the rectangular Tri-View may be changed. The Tri-View does not show perspective.

The geometry of surfaces is defined by vertices, edges and faces.

A vertex is a single point in space. It may be selected or unselected. Vertices are never visible in the images, but are displayed in the Tri-View either as purple dots for unselected vertices or yellow dots for selected vertices. A vertex may be 'indicated' by placing the cursor on it in the current window of the Tri-View. An unselected vertex can be selected by double-clicking on it. A selected vertex can be deselected by double-clicking on it . Vertices may also be selected if they are connected, by a network of edges, to an indicated vertex when SAJr. is commanded to SELECT CONNECTED.

An edge may be created between two vertices. An edge is displayed in the Tri-View as a dark line. Edges are not displayed in images, except in wire frame mode. An edge is considered to be selected if both its vertices are selected. An edge may be 'indicated' by placing the cursor on the edge in the current window, provided that the cursor is not also close to a vertex.

A face is created whenever three vertices are joined by three edges so each face is triangular. A face is considered to be selected if all three of its vertices are selected. Faces are not explicitly shown in the Tri-View but are displayed in

the images generated by SAJr., except in wire frame modes. Each face has a specified color. The color is defined by the reflection coefficients for red (R), green (G), and blue (B) light. The RGB coefficients range from zero to one hundred percent in 256 equal steps. The larger each RGB coefficient is, the more of that color light it reflects, and so, the more the selected face is tinted by that color.

The other elements that may be contained in the SAJr. universe are the ground, the sky, light sources, and the observer.

The ground, if it exists, is a plane that passes through the center of the SAJr. universe. It is displayed as a dark horizontal line through the origin of two of the Tri-View windows and is potentially visible in all imaging modes. The ground has a specified color with an RGB coefficient.

The sky, if it exists, is the background color of the SAJr. universe. The sky is not explicitly shown in the Tri-View but is displayed in the images generated by SAJr., except in wire frame modes. The sky has a specified color with an RGB coefficient.

There are two types of light sources in SAJr., background illumination and lamps. Neither illumination nor lamps are explicitly visible in images, but both have RGB coefficients and tint objects in images.

The background illumination is an ambient light source. It permeates the entire SAJr. universe completely in all directions and from all different directions.

A lamp is a point light source. Light flows from its position in space in all directions and become progressively attenuated as the distance increases. The position of a lamp is shown as a small white star in the Tri-View. A lamp may be 'indicated' by placing the cursor on the lamp in the current window. A lamp may be grabbed and moved while indicated, provided that no vertex is selected. In addition to having an RGB coefficient, lamps also have a brightness value.

The location of the observer in the Tri-View is shown by a small blue circle. The position of the observer's target is shown by a small blue x.

The Tri-View Gadgets

GADGET MODIFIER KEYS
gadgets:modifier keys

The effects of some of the following gadgets can be modified by holding one of the keyboard keys while the gadget is clicked. From left to right, these modifiers keys are: left shift, left ALT, left Amiga, right Amiga, right ALT, and right shift. The shift key is equivalent to its neighboring Amiga key. The left keys generally diminish the effect of a gadget, while the right keys enhance the effect. The ALT key has a greater effect on the command than the Amiga key. For some gadgets, pressing an ALT key at the same time as the neighboring Amiga key has a greater effect than either key singly. The degree of amplification or attenuation has been tailored to the fit the gadget's function and not all modifier keys are active on all gadgets.

DRAG BAR GADGET

This is a standard Intuition gadget located in the upper border of each window. It is used to drag the window to a new location on the screen.

THE BACK GADGET AND THE FRONT GADGET

These are standard Intuition gadgets that change the depth arrangement of windows on the screen.



THE REVERSE GADGET

This reverses the viewing direction of a Tri-View window. The first window is normally a downward view. Pressing this gadget will make it an upward view. This is achieved by replacing the contents of the window by its mirror image, thus the first window will have east on the left and west on the right.

THE MOVE TRI-VIEW GADGETS

These gadgets are located in the center of each border and cause the Tri-View to move in the direction of the gadget's arrows. The effect is to make the contents of the Tri-View move in the opposite direction. Holding down the left Amiga key while using a Move Gadget causes the size of the move to be smaller than usual. The left ALT key makes the displacement even smaller. If the right Amiga key is depressed, the Move Gadget moves the Tri-View further than usual. The right ALT key makes the displacement even larger.



THE EXPAND TRI-VIEW GADGET

This gadget causes the Tri-View to expand. If it is used with the left Amiga key, the expansion is small. If it is used with the right Amiga key, the expansion is large. The expansion can be made larger or smaller still by using the left or right ALT keys instead of the Amiga keys.



THE SIZING GADGET

This is a standard Intuition gadget for changing the size of a Tri-View window. Since each window represents a block of space, changing one window's size will result in one or more other windows changing size also.



THE CONTRACT TRI-VIEW GADGET

This gadget causes the Tri-View to contract. If it is used with the left Amiga key, the contraction is small. If it is used with the right Amiga key, the contraction is large. The contraction can be made larger or smaller still by using the left or right ALT keys instead of the Amiga keys.



THE CENTER TRI-VIEW GADGET

This gadget causes the Tri-View to move so that the cursor is left in the center of the Tri-View. Modifier keys: If the gadget is used while the left Amiga key is held down, the Tri-View windows are re-sized to their default size. If the gadget is used in conjunction with the right Amiga key, the Tri-View is centered on the default origin.



THE GRABBER GADGET

This gadget can be toggled on or off. If the Grabber is on, all the selected vertices will move when the cursor is moved. The Grabber is always turned off when the right mouse key is pressed to pull down a menu. When the Grabber is active, the icon for the cursor shows a small Grabber tool on the upper crosshair. If the Grabber is used when no vertices are selected, and the cursor indicates a lamp in the current window, then the Grabber can be used to move the lamp. If the Grabber is being used to move vertices, and all the selected vertices are within the Tri-View, then the effect of the Grabber can be undone by the Select gadget.



THE EDGE BUILDER GADGET

If either two or three vertices are selected, then edges are constructed between the selected vertices. If nothing happens when you use this gadget, except that the screen flashes at you, it means that there are more than three selected vertices. Perhaps the extra vertices are outside the Tri-View, or, you may have two vertices occupying the same position, both selected.



THE SELECT GADGET

This gadget changes the selection state of all vertices within the Tri-View. If one or more vertices in the Tri-View are selected, then they are unselected. If no vertices in the Tri-View are selected, then all the vertices in the Tri-View are selected. In either case, the state of vertices outside the Tri-View are unchanged.



THE ROTATE GADGETS

These gadgets cause selected vertices to rotate about the cursor in the indicated direction. Normally the vertices are rotated by 5 degrees for each click on the gadget. Modifier keys: If the gadget is used with the left ALT key the rotation is 0.1 degrees. If the left Amiga key is held down, then the rotation is by 1 degree. If the right Amiga key is held down then the rotation is by 45 degrees. If it is used with the right ALT key, the rotation is 180 degrees.

Non-standard Tri-View Mouse Usage

When the left mouse button is pressed down, and the pointer is over a Tri-View window, then the current tool is

dragged by the mouse. Special actions take place when a specific tool is picked, see the descriptions for TOOLS/CURVE, TOOLS/EXTRUDE, TOOLS/EDGE MAKER and TOOLS/GRABBER for more details. When the default tool is active, the cursor is represented by a blue cross and the following actions happen:

If the left mouse button is pressed, the cursor is moved to the pointer position. If the left button is held down while the mouse is moved, then the cursor will move.

If the right mouse button is pushed while the left button is held down, then a new vertex is created at the cursor position.

If the cursor is moved very close to a vertex in one window of the Tri-View, and the left mouse button is double clicked, the vertex will be selected if it was unselected, or vice versa. The cursor will snap to the location of the vertex in the other two windows.

If an image is being displayed on the screen, it may be pushed behind the Tri-View screen by first clicking on the image. The pointer will disappear. If the right mouse button is pressed, the image will be pushed behind the Tri-View screen.

Menu Commands

The menu commands are shown below. Each command is broken into two or three parts with a slash character (/). The first part is the menu heading, the second part is the menu item, and the third part, if present, is the submenu item.

PROJECT/LOAD/SCENE

This command invokes a file requester that displays the names of files that contain scenes. See the description of file requesters. If a scene file is selected, SAJr. opens the file and a requester pops up asking which portions of the file are to be loaded. A scene may contain data on objects, lamps, the observer or the world. Click on the boxes to toggle them between YES and NO to specify which parts of the file are to be loaded. Click on the OK button to start loading. A bow is also available to let you specify if you want objects in the scene to be loaded in a selected or unselected state.

Any objects or lamps that are loaded will be added, preexisting lamps or objects will be retained. If an observer or world is loaded, these attributes will override any existing observer or world attributes.

PROJECT/LOAD/IMAGE

This command invokes a file requester and displays the names of files that contain images. If a file is selected, the image is displayed.

PROJECT/LOAD/OBJECT

This command invokes a file requester and displays the names of files that contain scenes. If a file is selected, any objects that are found in the file are loaded at a position relative to the cursor.

PROJECT/LOAD/TAKE

A file requester is invoked that lets you select a Take.

PROJECT/LOAD/CODE

If this command is executed, an attempt will be made to keep all the program code in memory. If free memory falls below a prescribed minimum, then some code will be unloaded.

PROJECT/LOAD/WORKBENCH

An attempt will be made to load the Amiga Workbench.

PROJECT/SAVE/SCENE

This command causes the current scene to be written to the disk. A file requester pops up to solicit the name of a file. If a file name is chosen, a requester will then pop up asking which parts of a scene are to be saved. You may click on the boxes to toggle them from YES to NO. In addition you may specify if all objects are to be saved, or only the selected objects. Click on the box to toggle it between ALL and SELECTED. If the OK gadget is selected, the specified parts of the scene will be written to the file that was named earlier.

PROJECT/SAVE/IMAGE

This command is only available if an image has been rendered or loaded. It causes the current image to be written to the disk. A requester pops up to solicit the name of a file.

PROJECT/SAVE/OBJECT

This command causes selected objects in the current scene to be written to the disk. A requester pops up to solicit the name of a file. When the objects are written to the disk, their positions are taken to be relative to the cursor.

PROJECT/SHOW/IMAGE

If an image is loaded, then it will be displayed on the screen. The image can be pushed back by pointing at the image and first clicking with the left mouse button and then clicking with the right mouse button.

PROJECT/SHOW/PREVIEW

If a preview of an animation is loaded into memory, it is displayed. The preview can be stopped by clicking with the right mouse button or by pressing the escape key. The speed of the animation can be changed by pressing the keys 0 to 9. When a preview is stopped, you will be asked if you wish to unload it.

PROJECT/UNLOAD/IMAGE

If an image is loaded, it is removed, and the memory that it used is made available for other purposes.

PROJECT/UNLOAD/PREVIEW

If a preview is loaded, it is removed, and the memory that it used is made available for other purposes.

PROJECT/UNLOAD/CODE

All unneeded code is removed from memory until it is needed. As soon as code is no longer needed, it is again unloaded.

PROJECT/UNLOAD/WORKBENCH

An attempt is made to remove the Amiga Workbench. This will fail if a CLI window is present.

PROJECT/ABOUT

The version number and other information about SAJr. is displayed.

PROJECT/QUIT

After verifying that this is your intention, SAJr. stops operating and returns control to the workbench or CLI.

EDIT/SELECT/ALL

All vertices, and hence all edges and faces, become selected.

EDIT/SELECT/CONNECTED

All vertices that are connected, directly or indirectly, to the indicated vertex become selected. If a vertex lies within a few pixels of the cursor in the active window of the Tri-View, this vertex is considered the indicated vertex. If more than one vertex meets this criteria and they are distributed in depth, then the indicated vertex is the one that is closest to the cursor in the depth direction.

EDIT/SELECT/INDICATED VERTEX

The vertex that is indicated by the cursor is selected.

EDIT/SELECT/SWAP

All selected vertices become deselected and vice versa.

EDIT/SELECT/INDICATED EDGE

If an edge is indicated by the cursor, then the two vertices that define the edge are selected.

EDIT/DESELECT/ALL

All vertices, and hence all edges and faces become deselected.

EDIT/DESELECT/CONNECTED

All vertices that are connected, directly or indirectly, to the indicated vertex become deselected. If a vertex lies within a few pixels of the cursor in the active window of the Tri-View, this vertex is considered the indicated vertex. If more than one vertex meets this criteria and they are distributed in depth, then the indicated vertex is the one that is closest to the cursor in the depth direction.

EDIT/DESELECT/INDICATED VERTEX

The vertex that is indicated by the cursor is selected.

EDIT/DESELECT/SWAP

All selected vertices become deselected and vice versa.

EDIT/DESELECT/INDICATED EDGE

If an edge is indicated by the cursor, then the two vertices that define the edge are deselected.

EDIT/ERASE/SELECTED VERTICES

All selected vertices are erased. Any edges and faces that are defined by these vertices are also erased.

EDIT/ERASE/SELECTED EDGES

All selected edges are erased. Any faces that are defined by these edges are also erased. A selected edge is one that has a selected vertex at each end.

EDIT/ERASE/INDICATED VERTEX

The vertex that is indicated by the cursor is erased.

EDIT/ERASE/INDICATED EDGE

The edge that is indicated by the cursor is erased.

EDIT/ERASE/INDICATED LAMP

The indicated lamp is erased. If a lamp lies within a few pixels of the cursor in the active window of the Tri-View, this lamp is considered the indicated lamp. If more than one lamp meets this criteria and they are distributed in depth, then the indicated lamp is the one that is closest to the cursor in the depth direction.

EDIT/ERASE/ALL LAMPS

All lamps are erased.

EDIT/ERASE/ALL

All vertices, edges and lamps are erased. If a take is loaded, you will be asked if you want to have the take

erased. Erasing a take only removes the take in memory. The copy of the take on disk still remains.

EDIT/MODIFY/FACES

The modify faces requester is invoked. It can be used to change the default color, for yet to be created faces. It can also examine and change the color of selected faces.

EDIT/MODIFY/LAMPS

The modify lamps requester is invoked. It can be used to change the default color and brightness that is used for newly created lamps. It can also examine and change the color and brightness of either an indicated lamp or all existing lamps.

EDIT/MODIFY/WIRE-FRAME COLORS

This command invokes a color requester that can be used to modify the background color and the color used to draw edges for wire frame and preview renderings. Click on the button at the top right to toggle between FIELD and LINES, and use the color sliders in the usual manner.

EDIT/MODIFY/TAKE

If a Take is loaded, this command invokes the Modify Take requester, see below. This requester acts as a control panel for creating, modifying and rendering animations.

EDIT/DO/EXPAND

A requester is invoked that can be used to expand the set of selected vertices.

EDIT/DO/SUBDIVIDE

This operation divides each selected edge into two. Additional edges are created to ensure that subdivided faces remain triangulated. The effect is that faces that are selected are split into four, while faces that have two selected vertices are divided into two.

If only two vertices are selected, and they are not joined by an edge, then a requester pops up asking how many vertices to insert. Vertices are uniformly spaced between the two selected vertices and all are linked by edges.

EDIT/DO/SPIN

The spin operation takes place in the active Tri-View window. Each selected edge is swept around the spin axis creating new faces. Each selected vertex is swept around the spin axis creating new edges. The spin axis is horizontal on the screen and passes through the cursor position.

Normally the spin operation is applied to a sequence of connected edges created with the curve tool. It then produces a surface of revolution about the horizontal axis.

You are first asked how many steps are required for the spin operation. Then you are asked for a sweep direction.

EDIT/DO/REFLECT

The selected vertices and edges are duplicated by reflecting them in a plane defined to be vertical and to pass through the cursor location in the current Tri-View window. Any vertices in the plane of reflection are not duplicated.

EDIT/DO/MAKE TRI-VIEW SMALL

This command resizes the Tri-View windows to their default size.

EDIT/DO/MAKE TRI-VIEW BIG

This command makes the Tri-View windows big enough that they almost fill the screen.

EDIT/ADD/DUPLICATE

Each selected vertex and edge is duplicated. The duplicates are created in an unselected state. The duplicates occupy the same positions as the original, so they are not visible unless the Grabber is used to move the originals to a new location. This command may be used repeatedly as the original object is dragged from one location to another, in order to create multiple copies of the object.

EDIT/ADD/SPHERE

Add a pseudo-spherical polyhedron. A requester asks how many subdivisions of an icosahedron are required. A value of zero yields a regular icosahedron with 20 faces. Each extra subdivision quadruples the number of faces, so storage can easily be exceeded.

The size of the sphere is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/HEMISPHERE

Add a polyhedron shaped like a solid hemisphere. A requester asks how many subdivisions are required. The larger the value, the more closely the object resembles a hemisphere.

The size of the hemisphere is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/CUBE

A cube is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/PRISM

A prism is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/DISK

A disk is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/CIRCLE

A circle is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/CYLINDER

A cylinder is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/TUBE

A tube is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/CONE

A cone is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/LAMP

A lamp is created at the cursor location with the current brightness and color.

EDIT/ADD/VERTEX

A vertex is created at the cursor position. The vertex is selected.

EDIT/ADD/EDGES

This command has the same effect as clicking on the edge builder gadget.

TOOLS/CURVE

The curve tool resembles the cursor crosshairs except that the upper and right portions of the cursor are replaced by an icon of a pencil. When the curve tool is picked and the left mouse button is held down and the right mouse button is clicked, a new vertex is created and an edge is constructed from this vertex to the previous vertex, except if this is the first vertex made with this invocation of the curve tool. Prior to creating an edge, a rubber band line is drawn from the previous vertex to the cursor so as to indicate where an edge could be created. If the curve tool is indicating an existing vertex when the right button is clicked, then the previous vertex will be joined to the indicated vertex with an edge. The tool may be dropped by pressing the right mouse button or by the act of building an edge to an indicated vertex.

TOOLS/EXTRUDE

The extrude tool duplicates the selected object and joins the duplicate and the selected object. Edges are linked to create new faces only if the edge is a part of not more than

one face. This means that 'interior' edges do not extrude into faces. The Grabber tool is activated. If you move to another window and move the cursor, the selected object is dragged, while the duplicate is fixed. The effect is to extrude a two dimensional shape into a three dimensional solid. Remember to drop the Grabber when the extrusion is complete.

TOOLS/EDGE MAKER

The edge maker is used to join existing vertices with edges. A vertex is marked by moving the cursor to the vertex and while holding down the left mouse button, clicking with the right mouse button. The first and second marked vertices are joined by and edge, the third and fourth, etc. Prior to creating an edge, a rubber band line joins the previous vertex to the cursor to indicate the location that an edge would have.

TOOLS/GRABBER

This command has the same effect as clicking on the Grabber gadget in a Tri-View window.

OBSERVER/MODE/WIRE FRAME

This command sets the observer in wire frame mode. Rendered images only show edges and not faces. The perspective is correct.

OBSERVER/MODE/SKETCH

This command sets the observer in SKETCH mode. Sketch mode is similar to scanline mode, described below, except that rendering is faster and intersecting faces are not treated correctly. Sometimes a face is drawn in front of another face when it should be drawn behind.

OBSERVER/MODE/SCANLINE

This sets the observer in SCANLINE mode. Scanline mode is a variation of a 'painting' technique where each face is represented by single color and no shading is portrayed across a face. The image will be rendered more quickly for complicated scenes.

OBSERVER/MODE/LO-RES

This specifies that images are to made in Amiga low resolution mode with approximately 320 pixels across the screen.

OBSERVER/MODE/HI-RES

This specifies that images are to be made in Amiga high resolution mode with approximately 640 pixels across the screen. If Hold and Modify (HAM) graphics are employed, low resolution is used.

OBSERVER/MODE/NO-INTERLACE

This command specifies that images are to be produced without interlace, so that approximately 200 pixels are used in the vertical direction.

OBSERVER/MODE/INTERLACE

This command specifies that images are to be produced with interlace, so that approximately 400 pixels are used in the vertical direction. Some screen flicker may be perceived with a short persistence monitor.

OBSERVER/LOCATION

The location of the observer is moved to the cursor position. The observer's location is portrayed in the Tri-View by a small blue circle.

OBSERVER/TARGET

The observer's target is moved to the cursor position. The observer's target is portrayed in the Tri-View by a small blue x. The target position will be in the center of any images that are produced.

OBSERVER/LENS/NORMAL

This command sets the width of view to correspond to that of a 50mm lens on a 35mm camera.

OBSERVER/LENS/WIDEANGLE

This command sets the width of view to correspond to that of a 28mm lens on a 35mm camera.

OBSERVER/LENS/TELEPHOTO

This command sets the width of view to correspond to that of a 135mm lens on a 35mm camera.

OBSERVER/LENS/SPECIAL

This will invoke a requester for the focal length of a non-standard lens. Enter the value in millimeters. A small value, like 20, corresponds to a wide view, whereas a large number, like 100, corresponds to a narrow view where distant objects seem closer.

OBSERVER/EXPOSURE/AUTO

This command instructs SAJr. to determine the screen brightness automatically.

OBSERVER/EXPOSURE/MANUAL

This command overrides the automatic exposure setting. Numbers larger than 100 cause the image to be brighter, values less than 100 cause the image to be darker.

OBSERVER/IMAGE SIZE/MEDIUM

This command causes subsequent images to be made that are 1/2 of the full screen size.

OBSERVER/IMAGE SIZE/FULL

This command causes subsequent images to be made that are the full screen size.

OBSERVER/IMAGE SIZE/OVERSCAN

This command causes subsequent images to be made that are larger than the normal full screen size.

OBSERVER/START

This command initiates image generation. When an image is being generated, the grinding gear wheel icon vanishes after a short time and all SAJr. commands (except OBSERVER/START) may be executed without changing the image as it is produced. The image screen can be pushed behind the Tri-View screen by clicking on the image first with the left button and then with the right button.

OBSERVER/ABORT

This command will terminate any image generation.

OBSERVER/STATUS

This command will enable or disable the status line at the top of the screen. It shows the following information.

The amount of unused chip memory	Mem(**C**+F)
The amount of unused fast memory	Mem(C+**F**)
The number of selected vertices	V = **S** + U
The number of unselected vertices	V = S + **U**
The number of edges	E =
The number of faces	F =
The number of lamps	L =

WORLD/SKY

This command invokes a sky requester, which closely resembles the color requester. The gadget at the top is a two way toggle that can take the values NONE or SOLID. Click on the gadget to change its value. When NONE is selected, no sky is generated, and the color sliders are not used. If SOLID is selected, then a single color sky is produced, and the color sliders can be used to specify the color.

WORLD/GROUND

This command invokes a ground requester, which closely resembles the color requester. The gadget at the top is a two way toggle that can take the values NONE or SOLID. Click on the gadget to change its value. When NONE is selected, no ground is generated, and the color sliders are not used. If SOLID is selected, then a ground

plane is created and the color sliders can be used to specify the ground color.

If SOLID is chosen, then a line representing the ground is drawn in two of the Tri-View windows.

WORLD/ILLUMINATION

This command invokes a color requester that can be used to set the brightness and color for background illumination. This light source bathes the entire scene without casting shadows. Normally the V value should be set at about 1/4 to prevent shadows from being totally black. Too high a value, or the absence of lamps will result in a flat and lifeless image.

The File Requester

This requester is used to select new or existing files. It has three main panels. The panel on the upper right lists the currently mounted disk volumes. To select a new volume, double click on the volume that you need. If more than six volumes are present, the scroll bar or the scroll gadgets can be used. The currently selected volume is highlighted.

The panel on the upper left works in a similar manner, but shows the drawers on the current volume. Either double click on drawer, or type in the name in the box immediately above the panel. The first item in the panel is marked '/ Parent Directory' and is used to select the drawer containing the current list of drawers. Using this panel you can traverse a set of nested drawers.

The panel on the lower left is used to select files once the volume and drawer have been selected. Again double click on the file or enter a name in the box above the panel. To create a new file, of course, you must enter its name.

Additional buttons allow you to rename or delete files. The OK button can be used to select a file, if the desired file has been highlighted by clicking on it.

The Expansion Requester

This requester can be used to expand the set of selected vertices. The expansion (or contraction) is uniform in the selected directions and is directed away from the cursor.

The expansion can take place in the north-south direction or the east-west direction, or the up-down direction, or any combination selected by the buttons on the requester. Expansion may be in discrete stages by factors of 2 or 1/2 by pushing the appropriate buttons. Alternatively the upper slider may be used. Moving it to the right of center causes expansion, left of center causes contraction. The lower slider may be used for fine adjustment. It is 50 times less sensitive.

The expansion requester may be dragged to a new location so that it does not obstruct the Tri-View. The drag bar is at the top of the requester. If the CANCEL button is pressed, the vertices revert to their positions before the expansion was attempted.

Color Requesters

Several requesters allow you to select a color. The color may be changed by using sliders to set the red, green and blue values. Alternatively, sliders can control the hue, saturation and value. Hue varies continuously from red to green to blue and back to red. Saturation represents the purity of the color. As the saturation is decreased, more white is mixed in. The value is a measure of the brightness of the shade. The color represented by the sliders is displayed on the color swatch at the bottom of the requester.

The Modify Faces Requester

This requester is used to modify the current face color. The color sliders work as described above.

The colors of selected faces can be modified by pressing the SET gadget. The colors of selected faces can also be extracted, if the colors of all selected faces are the same, with the FETCH gadgets. An average color can be extracted even when the colors of selected faces differ, by the BLEND gadget.

The Modify Lamps Requester

This requester is used to modify the current color and brightness of lamps. If a lamp is indicated by the cursor, then the gadget marked WHICH LAMP can be toggled between ALL, meaning that all lamps are to be considered, and INDICATED which means that only the indicated lamp is to be considered.

Colors can be set in the usual way with the sliders. FETCH and BLEND gadgets are provided for extracting the color and brightness, while SET gadgets can be used to change the properties of lamps.

The Modify Take Requester

This requester can be used to change the parameters of the Take and to initiate various actions. The gadgets that are available, in some cases depends upon the settings of other gadgets. The following gadgets are available:

Take mode:

This gadget is always available, it can be toggled between FRAME and GLOBAL. When it is set to GLOBAL, only gadgets that relate to the animation as a whole will be accessible. When it is set to FRAME, then

only gadgets that are relevant to a particular frame will be available.

Number of frames:

This gadget can be used to enter the total number of frames that are needed by the animation.

RAM animation:

This gadget can be cycled through REGULAR, ECONOMY.and ANIM-5. The ANIM-5 results in the fastest compression, while REGULAR usually produces the fastest playback.

Loop mode:

This gadget can by cycled though NONE, LOOP and OSCILLATE. This gadget is only available when the RAM animation gadget is not set to NO. If NONE is specified, then any resulting animation will play once and then stop. If LOOP is requested, then the animation will be repeated until it is stopped manually. If OSCILLATE is set, the animation will first be played forwards and then backward. This sequence is repeated until the animation is stopped. Loop control does not apply when the ANIM-5 style of animation is used.

Preview size:

This gadget can be toggled between FULL and HALF and it controls the size of the wire frame preview display, when the PREVIEW button is pressed.

Save images:

This gadget can be toggled between YES and NO. Normally it is set to YES and all images that are rendered are saved to disk. If the gadget is set to NO, then the images may still be saved to disk, but they will be deleted as soon as possible.

SAVE GLOBAL SCENE

This gadget causes the objects in the Tri-View to be saved as the global scene for the Take.

LOAD GLOBAL SCENE

This gadget causes the Tri-View to be cleared, and the global scene for the Take to be loaded.

DELETE ALL IMAGES

This gadget deletes all the images associated with the Take. You should use this at any time that you have modified the Take in any that would make the rendered images invalid.

PREVIEW

This gadget starts a moving wire frame preview. The preview will be repeated when it is finished. The preview may be stopped at any time by pressing the right mouse button or the escape key. The speed can be controlled with the keys 0 to 9.

RENDER ALL

This gadget initiates the rendering process. When all the the images have been produced an animation file will be generated and saved to disk.

OK

This gadget permits you to leave the requester.

Current frame:

This gadget is only available when the 'Take mode' is set to FRAME. You may use this gadget to specify which frame number you are interested in. Key frame: This gadget can be toggled between YES and NO. It is used to specify whether the current frame is a key frame.

Duration:

This gadget can be used to specify the length of time, in jiffies, that the current frame should be displayed.

Image drawer:

This gadget can be used to specify a drawer for the current image. This gadget is particularly useful if you are using floppy disks and all the images will not fit on a single disk. If this item is left blank, and an image drawer is specified for an earlier frame, then that drawer will be used for the current frame.

Scene drawer:

This gadget can be used to specify a drawer for the key frame scene for the current frame. This gadget is particularly useful if you are using floppy disks and all the scenes will

not fit on a single disk. If this item is left blank, and a scene drawer is specified for an earlier frame, then that drawer will be used for the current frame.

SAVE KEY FRAME SCENE

The objects that are in the Tri-View are saved to become the key frame scene for this frame. Any objects that are derived from the global scene will not be saved.

LOAD KEY FRAME SCENE

The key frame scene will be loaded if this frame is a key frame. Objects from the global scene will also be loaded, but these will not be able to be modified. If the current frame is not a key frame, then objects will be loaded by in-betweening, but they will not be able to be modified, unless you specifically request it.

DELETE CURRENT IMAGE

The image associated with the current frame will be deleted.

RENDER FRAME

An image corresponding to the current frame will be rendered.

Notes

NOTES

APPENDIX B

GLOSSARY

AXES
The plural of axis.

AXIS
A principal direction that is used for measuring the position of an object. In SAJr., the axes are directed to the east, north and upwards.

CLICK
The expression 'click on a gadget', means that you are to move the mouse pointer to the gadget and press down the left mouse button and then release it. If you press the button, and then decide that you want to change your mind, some gadgets permit the command to be cancelled by moving the pointer away from the gadget before releasing the button.

CURSOR
This is an object that represents a position in three dimensions, so it is displayed in all three Tri-View windows. It can be moved by the mouse while the left button is pressed down.

CURVE
In SAJr., a curve is a series of vertices linked by edges. Each vertex on a curve must have either one or two edges associated with it.

DELTA COMPRESSION
A system of compacting a series of files based on saving only the differences between consecutive files.

DELTA DATA
A description of the difference between two images. Given one image and the appropriate delta data, then a second image can be reconstructed. If two images are very similar,

then the size of the delta data is much smaller than the size of the image.

DOUBLE BUFFERED ANIMATION

A method for displaying an animation using a computer. Each image is built in an off screen page of video memory and displayed only when complete. Two pages of video memory are used so that as one is being constructed, the other can be on display.

DOUBLE CLICK
Because the mouse only has two buttons, you can signal the computer that you wish to do something special, by pressing a button twice in rapid succession. This is called double clicking, and it is almost always done with the left mouse button. You can use the Workbench Preferences tool to set the interval of time that the Amiga regards as 'rapid succession'. See the book "Introduction to Amiga" for more details. With SAJr. you should set the interval to be fairly short, so that a double click is a very deliberate act, rather than normal frenzied mouse activity.

DRAG BAR
This is a gadget that is located in the top border of each window of the Tri-View. It is used to move the window around on the screen.

EDGE
In SAJr. an edge is a line joining two vertices. An edge is visible as a dark line in the Tri-View, but is never seen in a picture.

EDGE BUILDING GADGET
This gadget resides in the lower left border of each Tri-View window. Its icon is a triangle. If exactly two vertices are selected, then an edge will be constructed between the vertices. If three vertices are selected, then they will be joined by three new edges.

FACE
In SAJr. objects are built from a patchwork of triangular faces. A face has a particular color. A face is automatically created whenever three vertices are joined together by three edges.

FRAME
One image of an animation. Often a frame is made up by interlacing two fields, one containing odd numbered scan lines and the other containing even numbered scan lines.

GADGETS
These are small images that can be used to perform an operation. Some gadgets are common to most Amiga programs while others are unique to SAJr. The Tri-View has a number of custom gadgets in the borders of each window, that are used for changing the position and size of the Tri-View and for manipulating objects. Other gadgets appear in requesters. Most gadgets are operated by clicking on them with the left mouse button. Others, called sliders, are like the controls on a hi-fi graphic equalizer. You can click on the control knob and then slide it up and down, or from side to side, in order to adjust a value. Finally there are gadgets, that when selected by clicking, will enable you enter characters or a values from the keyboard.

HORIZON
The part of the sky that appears to be at ground level.

IN-BETWEENING
The process of animation interpolation, whereby the position and shape of an object at one time is inferred from its position and shape at other times. In-betweening, betweening, or tweening, automates motion from one specified state to another. The frames for which the scene is explicitly provided are called key frames.

INDICATED
Marking an object by placing the cursor on or near it in the current window of the Tri-View.

INDICATED VERTEX
If a vertex lies within a few pixels of the cursor in the active window of the Tri-View, this vertex is considered the indicated vertex. If more than one vertex meets this criteria and they are distributed in depth, then the indicated vertex is the one that is closest to the cursor in the depth direction.

JIFFY
The length of time that it takes to display a single field of an image. On NTSC systems this time is 1/60th of a second. On PAL systems the time is 1/50th of a second.

KEY FRAME
An animation frame in which all details of the scene are explicitly provided. The non-key frame scenes are interpolated from the two nearest key frames before and after. This process is called in-betweening.

LENS
The lens selects the width of view and perspective distortion.

LOCATION
The point from which a scene is viewed in order to render an image. The width of view can be controlled by choosing a lens type.

LOOP
In SAJr., a loop is a series of vertices linked by edges where each vertex is associated with two edges. It is a closed curve.

MODIFIER KEYS
Keyboard keys that, if held down while certain Tri-View gadgets are clicked, change the effect of a gadget.

OBSERVER
This entity embodies all the attributes required to specify the process of generating an image, including a viewing location, a target position, an imaging method and an image size. The width of view can be controlled by choosing a lens.

ORIGIN
The initial position of the cursor.

OSCILLATE
In SAJr., when an animation is said to oscillate, it runs the animation alternately forwards and backwards.

PIXEL
A picture element on the screen. The Amiga normally displays 320 or 640 pixels in the horizontal direction and 200 or 400 pixels in the vertical direction.

POINTER
This is the image that moves on the screen when you move the mouse. It looks like a red arrow, unless it has changed it with the Preferences tool.

PREVIEW
The process of rapidly generating a rough draft moving image of an animation. In SAJr., the preview is displayed in wire frame mode.

RAM ANIMATION
A method for displaying an animation using a computer. Each image is stored in memory in a compressed form and is expanded before being displayed.

RENDER
The word render is used to apply to the process that generates an image of a scene.

REQUESTER
This is an object that appears on the screen from time to time. It appears when SAJr. needs some extra information or wants to be sure that you have received a message before continuing. Requesters contain gadgets like push buttons, sliders, or even little boxes where you can type a short message. Whenever a requester pops up, you must acknowledge it by answering any questions and pressing the OK or CANCEL button with the left hand mouse button before continuing with SAJr.

RUBBER BAND LINE
A dotted line drawn from a fixed point to the cursor to show where an edge could be built.

SELECT
A vertex is either selected or deselected. The selection state of a vertex can be changed by double clicking on it, or by one of several menu commands. Many operation only apply to selected vertices.

SCENE
A scene is one of the file formats used by SAJr. It can contain objects, lamps, an observer,and a world. In order to be recognized as a scene, the file name should end with '.scene'. The command PROJECT SAVE SCENE automatically adds this suffix.

SNAP
If the left mouse button is double clicked in a window, SAJr. looks for a nearby vertex (within about 3 pixels). If more than one pixel satisfies the criteria, then the one that is closest in the 'depth direction' of the window is chosen. The selection state of the vertex is toggled. Also the cursor is moved to the position of the vertex in the direction perpendicular to the plane of the current window. This is called snap.

SOLID OBJECT MODELING
In this manual, solid object modeling is defined as the creation and manipulation of three dimensional mathematical facsimiles of objects, for the purpose of creating images.

SWATCH
The patch of color at the lower center of one of the color requesters is called a swatch. It indicates the current color. Move the sliders to change the color.

TAKE
A sequence of frames that make up an animation. In SAJr., a Take is a file that contains enough information that the image for each frame can be rendered, and the images assembled into an animation.

TARGET
A point within a scene that will appear in the center of a rendered image. The point toward which the observer always looks.

TOGGLE
When a gadget has more than one state of existence, and a single action causes it to change from one to another, it is said to toggle. For example: some requesters have a box that may either contain the word YES or the word NO. Clicking on the gadget will change the gadget from YES to NO or vice versa.

TOOL
A tool allows you to interactively manipulate objects. In SAJr., the active tool is indicated with the cursor. The default tool, i.e. the one represented by the cross-hair cursor, is used for indicating things and for creating vertices. The grabber gadget changes the default tool into the grabber tool which is used for moving selected objects. Other tools include the the curve tool, for drawing freehand curves, the extrude tool, for manipulating extrusions, and the edge maker tool, for making edges.

TRI-VIEW
The set of three windows that are used as a workspace for manipulating three dimensional objects. The Tri-View represents a block of three dimensional space, and only objects within that block are displayed. Objects are displayed in wire frame form in the Tri-View.

VERTEX
This is a point in space that can define the end of an edge, or the corner of a face. In SAJr., you manipulate an object by changing the position of one or more vertices.

VERTICES
The plural of vertex.

WINDOW
A portion of the screen that sometimes can be moved around or rearranged in depth. The most important windows in

SAJr. are the three windows of the Tri-View. These windows have blue and white border areas containing a plethora of gadgets. The interiors of the Tri-View windows are brown and contain wire frame renditions of the contents of the Tri-View.

Notes

NOTES

INDEX

ABORT 3-11, A-23
ABOUT A-11
ADD
    circle 5-3, A-17
    cone 5-3, A-17
    cube 5-4, A-17
    cylinder 5-3, A-17
    disk 5-3, A-17
    duplicate 6-8, A-16
    edges 4-3, A-18
    hemisphere 5-3, A-16
    lamp 2-23, 4-5, 7-1, A-18
    prism 5-5, A-17
    sizing 5-4
    sphere 5-1, A-16
    tube 5-3, A-17
    vertex 4-2, A-18
ALT key 2-19, 6-4
Amiga key 2-18, 6-4
ANIM format
    See Take RAM animation
animation
  See Chapters 8 and 9
    betweening 8-6
    cel animation 8-6
    exposure 9-9
    key framing 8-6
    lens 9-9
animations 1-15
AXIS B-1
Back gadget 2-25
background 7-5
background illumination A-2
backup 2-1
Betweening 8-6, 8-8
    distortion 8-11 (illus.)
Birth Order 8-6, 8-8
blend color 4-14
Byte by Byte Corporation
    Foreword-iv
CANCEL 2-11
cardinal direction 2-4, 2-15
Center Tri-View gadget 2-19
chip memory 3-11, 3-12
CLICK B-1
Cntrl-C 8-22
color 4-12, A-25
    blend 4-14
    coefficients A-2
    fetch 4-14
    map lock 9-8
    reflection A-2
    requester 4-12, A-25
    RGB A-2
    Swatch 7-5, A-25
Commodore key 2-18
contract A-25
Contract Tri-View gadget 2-20
Creating Takes 8-2
Current frame 8-4
cursor 2-6, 2-19, B-1
curve B-1
    tool 5-6, A-18
DELETE
    ALL IMAGES 8-20, A-28
    CURRENT IMAGE
        8-20, A-30
    HIGHLIGHTED FILE 2-11
DELTA
    COMPRESSION B-1
    DATA B-1
deselect
    all 4-9, A-12
    connected 4-9, A-12
    indicated
        edge 4-10, A-12
        vertex 4-9, A-12
    swap A-12
    vertex A-1
disk space 9-4
DOUBLE BUFFERED
ANIMATION B-2
Double buffering 8-21
DOUBLE CLICK B-2
DRAG BAR B-2
drawer 2-10
    image A-29
    scene A-29
drive 2-10
DUPLICATE 6-8

Duration 8-4, A-29
jiffy 8-4
edge 1-12, 4-2, A-1, B-2
count 3-13
Builder gadget
4-2, 4-6, B-2
Maker tool 5-7, A-19
EDIT See *submenu items*
DO See *submenu items*
ERASE
all 4-3, A-13
lamps 4-11, A-13
indicated
edge 4-11, A-13
lamp 4-11, A-13
vertex 4-11, A-13
selected
edges 4-11, A-13
vertices 4-10, A-13
ERASE ALL 4-11
ESC 2-9
EXPAND 6-6, A-14, A-25
requester A-25,
(.illus) 6-5
Expand Tri-View gadget 2-18
EXPOSURE 3-7, 7-3, A-22
lock 9-8
Extrude 1-13, 5-10, 6-9
triangulation 5-11
EXTRUDE tool A-18
face 1-11, 4-4, A-1, B-3
count 3-13
diagram 4-4
fast memory 3-11, 3-12
fetch
color 4-14
file name
Conventions 9-3
extensions 2-24
File Requester 2-10, 2-24
(See illus.) 2-11
Film Strip gadget 8-6, 8-13
finding objects 2-20
floppy disk 2-10
FRAME B-3
frame drawer 9-6
frame rate (table) 8-15
gadgets B-3
Film Strip (See illus.) 8-4
intuition
Back 2-5
Depth Arrangement 2-5
Drag Bar 2-5
Front 2-5
Sizing 2-5
modifier keys A-3
Requester
See specific requesters
text entry 2-24
toggle 7-5
Tri-View
See Tri-View gadgets
Center 2-19
Contract Tri-View 2-20
Edge Builder 4-2, 4-6
Expand 2-18
Grabber 4-15, 5-4
Move Tri-View 2-16
Reverse 2-21
Rotate 6-4
Select 4-8

Global Scene 8-5
Grabber 4-15
gadget 5-4
techniques 4-17
tool A-19
undo 4-17
grinding gears 2-9, 3-7
ground 1-13, 7-6, A-2, A-23
requester (See illus.) 7-7
hard disk 2-10
hidden faces 4-8
hide image 2-12
HORIZON B-3
hue 4-13
icons 2-24
IFF 1-15, 2-24
ILBM 2-24
Illumination 7-4, A-24
requester (See illus.) 7-4

image 1-14, 1-15
drawer 9-6, A-29
hiding 2-12, 2-23
load 2-9
save 2-24
show 2-23
size 2-23, (table) 3-8
unload 2-12
Images
In-betweening
See *Betweening* B-3
Indicate 2-6
edge A-1
lamp A-2
vertex A-1
indicated B-3
vertex 4-6, B-4
jiffy *See duration* B-4
Key Frame 1-16, B-4
loading 8-7
saving 8-7
Key Frame animation 8-5
Key Frame gadget 8-5, 8-13
keyboard commands A-3
lamp 1-14, 2-22, 7-1, A-2
arrangement 7-3
brightness 7-1
color 7-1
erase 4-11
modify 7-1
move 4-16, 7-2, B-4
lathe See *spin*
left mouse button 2-7
lens 3-5, A-21, B-4
zoom 3-5
load
code 3-11, A-9
Global Scene 8-5, A-28
image 2-9, A-8
key frame 8-7,A-30
lamp A-8
object 6-3, A-8
scene 2-13, 6-2, A-8
Take 8-2, A-8
Workbench 3-12, A-9
world A-8
location 2-21, 2-22, 3-4, A-3,
See observer B-4
lock
color map 9-8
exposure 9-8
LOOP B-4
Loop mode 9-1
lozenge 5-5
make
Tri-View small 2-6, A-15
Tri-View big 2-6
memory 3-11
menu system 2-7
techniques 2-8
MODE 3-1, A-19
HI-RES A-20
INTERLACE A-20
LO-RES A-20
NO-INTERLACE A-20
SCANLINE
3-2, 6-11, A-20
SKETCH 3-2
WIRE FRAME 3-1, A-19
modifier keys 2-18, B-4
MODIFY
faces 4-12, A-14
Faces Requester
(See illus.) 4-12
lamps 7-1, A-14
Take 8-2, A-14
Wire Frame colors
3-2, A-14
mount 2-10
mouse
buttons 2-7
usage A-6
Move
lamp 4-16, 7-2
vertices 4-15
Tri-View gadget 2-16
movie 8-22
multiple menu commands 2-8
NTSC 3-8
Number of frames 8-3
objects 4-1

Observer
1-14, 2-13, 2-21, A-19, B-4
See Also *submenu items*
LOCATION A-21
target 3-5, A-21
OK 2-11
Origin 7-7, B-5
orthogonal view 4-8
orthographic 3-1
OSCILLATE 9-1, A-27, B-5
OVERSCAN A-22
paint programs 2-24, 3-7
PAL 3-8
palette See *color*
Parent Directory 2-10
PIXEL B-5
plumbing fixture 6-9
POINTER B-5
PREVIEW
8-15, A-27, A-28, B-5
primitives See *Add*
profiles 5-6
PROJECT See *submenu items*
QUIT 2-25, A-11
RAM animation 1-16, 9-9, B-5,
See *Take*
RAM disk 2-10
RAM-Animation *Introduction-x*
rectangular block 6-6
REFLECT 6-8, A-15
RENAME HIGHLIGHTED FILE
2-11
render B-5
all 8-19, A-29
frame 8-17, 8-20, A-30
requester
file A-24
Modify faces 4-12, A-26
lamps A-26
Take A-26
Reverse gadget 2-21
RGB A-2
right mouse button 2-7
Rotate gadget 6-4
rubber band line 5-6, 5-7, B-6
rule
Key Framing 8-10
Two Window 2-7
saturation 4-13
save
Global Scene 8-5, A-28
image 2-24, 9-7, A-9
key frame 8-7
KEY FRAME SCENE
8-6, A-30
lamp A-9
object 6-3, A-9
scene 2-24, 6-1, A-9
Take 8-4
world A-9
SCANLINE 3-2, 6-11
scene 1-13, 2-13, B-6
drawer A-29
load 2-13
save 2-24
screen resolution 3-3
scroll bar 2-12
Select B-6
all 4-9, A-11
connected
4-9, 4-16, 9-11,
A-1, A-11
edge A-1
face A-1
gadget 4-8
indicated
edge 4-10, A-12
vertex 4-9, A-11
swap 4-9, A-11
vertex A-1
selected 4-5
Shift 2-19
show
image 2-23, 3-10, A-10
preview 8-15, A-10
Sizing gadget 2-5, 2-18
SKETCH 3-2, A-19
sky 1-13, 7-5, A-2, A-23
Requester (See illus.) 7-5
slider 2-12
SNAP B-6

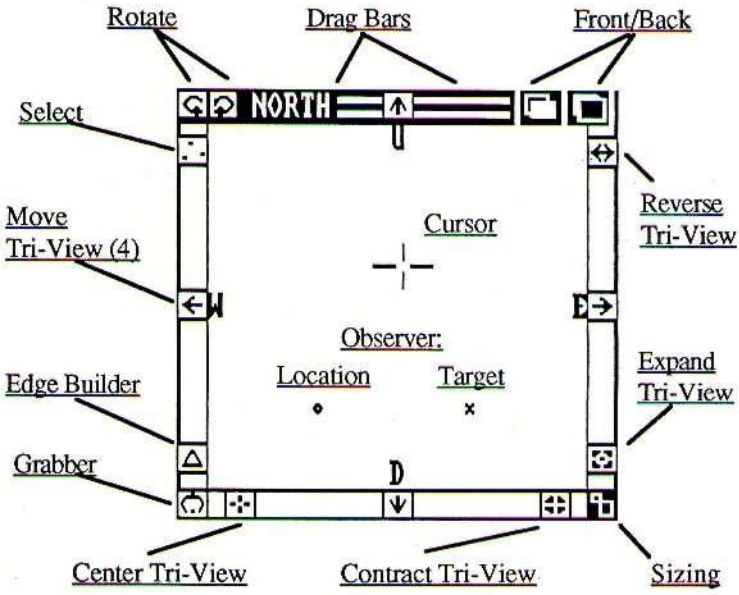
SOLID OBJECT MODELING
B-6
SPIN 5-8, A-15
(See illus.) 5-9
stack 2-1
START 2-23, 3-9, A-22
Starting SAJr 2-1
STATUS 3-11, 3-13, A-23
Stopping a render 8-22
SUBDIVIDE 5-1, 5-3, A-15
(See illus.) 5-2
techniques 5-3
submenu 2-8
items *(Actually, we were referring to each specific menu in this particular category.)*
support policy See *Foreword*
swatch 7-5, A-25, B-6
Take 1-15, 8-1, A-26, B-6
creating 8-2
FRAME 8-2, 8-3
Mode Requester
(See illus.) 8-4
GLOBAL 8-2
Mode Requester
(See illus.) 8-3
Global Scene 8-5
load 8-2
Global Scene 8-5
key frame 8-7
modify 8-2
RAM animation 8-18
ANIM-5, 8-21
ECONOMY 8-21
REGULAR 8-21
Render all
interrupting 8-22
save 8-4
Global Scene 8-5
key frame 8-7
TARGET 2-22, 3-5, A-3, B-7
See observer
text entry gadget 2-24
TOGGLE B-7
toggle gadget 7-5
TOOL B-7
Tools
See *submenu items*
Curve 5-6
Edge Maker 5-7
Grabber 4-19
Tri-View 1-12, 2-3, A-1, B-7
gadgets
Center A-5
Contract A-5
Edge Builder A-6
Expand A-4
Grabber A-5
Reverse A-4
Rotate A-6
Select A-6
Intuition
Back A-3
Depth Arrangement A-3
Drag Bar A-3
Front A-3
Sizing A-4
Move A-4
Make small A-15
triangulate 5-7
troubleshooting 2-26
Tweening *See Betweening*
Undo grabber 4-17
unload
code 2-2, 3-11, A-10
image 2-12, 3-10, A-10
Preview 8-16, A-10
Workbench 2-2, 3-12, A-10
unselected 4-5
value 4-13
vertex 1-12, 4-1, A-1, B-7
count 3-13
vertices 4-1, B-7
moving 4-15
volume 2-2, 2-10, 9-6
WINDOW B-7
wire frame 1-14, 2-14, 2-23, 3-1
working scene 8-7
World 6-2
See *submenu items*
zoom lens 3-5

NOTES

BYTE by BYTE™ CORPORATION

SCULPT-ANIMATE™ 4D Jr. QUICK REFERENCE CARD

Tri-View Gadgets



Gadget:	Purpose:
Rotate	Rotate selected vertices
Move Tri-View	Move Tri-View around universe
Reverse Tri-View	Look from opposite direction
Expand Tri-View	Make Tri-View show more (zoom out)
Contract Tri-View	Make Tri-View show less (zoom in)
Center Tri-View	Center Tri-View on cursor position
Grabber	Invoke the grabber tool
Edge Builder	Connect 2 or 3 selected vertices
Select	Select/Deselect Tri-View vertices *only*

Playing a RAM Animation

SAJr. automatically creates an icon for each animation. Simply double click the icon for the animation that you wish to play back. The **MOVIE** program must be present in the same directory as your animation icon.

Animations compressed with the ***ANIM5 option*** require a public domain program called **SHOWANIM**. Consult your local dealer or user group for a copy of this program.

Center Tri-View gadget modifier.

To return the TriView display to the original zoom scale, window size, and position; select the Center TriView gadget while pressing the Left Amiga key.

To center the TriView at the origin, which is coordinate (0,0,0); select the Center TriView gadget while pressing the Right Amiga key.

Observer/Status Bar (F10)

This command will enable or disable the status line at the top of the screen. It shows the following information.

The amount of unused chip memory	Mem(**C**+F)
The amount of unused fast memory	Mem(C+**F**)
The number of selected vertices	V = **S** + U
The number of unselected vertices	V = S + **U**
The number of edges	**E** =
The number of faces	**F** =
The number of lamps	**L** =

Technical Support (512) 343-4376

Registered SAJr. owners are entitled to 30 days (from date of purchase) of free technical support. Your SAJr. warranty card **must** be on file before any support can be given. Technical support may be purchased at $15 per ten minutes or at $75 per hour. Please do not call for technical assistance until you are sure that the manual does not answer your question. Technical support hours are: **(10am-5pm, CST)**

Keyboard Mouse Equivalence

The Amiga supports the use of the keyboard to emulate the mouse. Holding either Amiga key while pressing one of the "arrow" cursor keys will move the mouse pointer. Note that some Amiga keyboards may label the left Amiga key with the Commodore logo. When an Amiga key is being held, the left and right ALT keys emulate the two mouse buttons. This built-in feature lets you manipulate the Tri-View cursor without touching the mouse.

Keyboard Menu Equivalents

Pre-defined keyboard short-cuts are provided for a number of Sculpt Animate 4D Jr.'s menu commands. These are listed in the menus next to the commands they invoke. Pressing one of these key-sequences is the same as selecting the associated menu command.

To use an equivalent, follow one of these examples:

F10	Press the similarly marked "function" key at the top of your Amiga's keyboard
A D	Hold down the right-hand "Amiga" key and press the associated letter key.
^C	Hold down the "CTRL" key at the left end of the keyboard and press C.

Key	Command	Key	Command
F1	Load Scene	F6	Save Object
F2	Load Image		
F3	Load Object	F8	Observer Start
F4	Save Scene	F9	Show Image
F5	Save Image	F10	Status Display
A B	Edge Maker	A N	Add Vertex
A D	Subdivide	A O	Observer Location
A E	Erase Selected Vertices	A P	Deselect Indicated Vertex
A F	Modify Faces	A U	Curve Tool
A I	Erase Indicated Vertices	A X	Expand
A L	Add Lamp	A Y	Select Indicated Vertex
A M	Modify Take	^C	Abort

Using the Mouse

Mouse Action:	Effect:
Click Left Button	Move the cursor
Double-Click Left	Select indicated vertex
Click Left-Right	Hide displayed image or animation
Hold Left, Click Right	Create new vertex under cursor
Click Right	Drop the current tool
Hold Right	Use the menus

This behavior will vary if a tool has been selected. Except for menu use, these actions take effect only if the mouse pointer is positioned over one of the Tri-View windows.

Mouse Modifiers

The ALT and SHIFT keys at either side of the keyboard may be used to modify the behavior of certain gadgets and tools, making their action finer or coarser. The gadgets that can be modified are Rotate and Move Tri-View. The SHIFT keys will also modify the Expand and Contract Tri-View gadgets.

Modifier Key:	General Effect:	Rotation:
Left ALT	Very fine	0.1°
Left SHIFT	Fine	1°
(no modifier)	Normal	5°
Right SHIFT	Coarse	45°
Right ALT	Very Coarse	180°

The Left and Right Amiga keys have the same effect as the SHIFT keys in most circumstances, but they do not work the same when used with the cursor keys.

Move Tri-View Keyboard Equivalent

Whenever a Tri-View window is selected, the "arrow" cursor keys on the Amiga's keyboard may be used as a substitute for the Move Tri-View gadgets. The SHIFT and ALT keys will vary the amount of motion as usual.

Last Command Repeat

The **ESC** key at the upper-left corner of your Amiga's keyboard will automatically repeat the last menu command.