

$\alpha^2$

# AlphaMania<sup>TM</sup> 2

## User Manual

# Table of Contents

<b>PART 1: OVERVIEW</b>	<b>4</b>
If You Are An AlphaMania 1.0 User	4
What AlphaMania Does	4
AlphaMania Is Easy To Use	5
What's new in AlphaMania 2?	5
Migrating from AlphaMania 1 to AlphaMania 2	6
<b>PART 2: USERS GUIDE</b>	<b>7</b>
Operating Systems Supported	7
Installation	7
Registering & Unlocking AlphaMania	7
Importing PICT and Targa Files	8
Batch Importing	9
AlphaMania Basic Cast Properties	12
AlphaMania Basic Sprite Properties	14
Distributing AlphaMania 2 with Projectors & Shockwave	15
<b>PART 3: REFERENCE</b>	<b>16</b>
Anatomy of an Alpha Channel	16
Creating and Editing Alpha Channels	18
Techniques for Creating Alpha Channel Effects	19
<b>PART 4: OPTIMIZING ALPHAMANIA</b>	<b>25</b>
Speed Issues	25
Space Issues	25
Quality Issues	26
Limitations and known problems	26
<b>PART 5: FREQUENTLY ASKED QUESTIONS:</b>	<b>28</b>

<b>PART 6: ALPHAMANIA 2 AND SPECIAL EFFECTS</b>	<b>30</b>
Drawing Methods	30
Using Effects	36
<b>PART 7: THE SCALE EFFECT</b>	<b>46</b>
Scale Basics	46
Scale Reference	47
<b>PART 8: REGISTER, PURCHASE, DISTRIBUTE, CONTACT INFO</b>	<b>50</b>
How to Register AlphaMania	50
Purchasing	50
Distributing AlphaMania	51
Contacting Media Lab, Inc.	51
<b>PART 9: LEGAL STUFF</b>	<b>52</b>
Licensing	52
Copyright Information	52
Trademarks	52
License Agreement	52

# Part 1: Overview

## *If You Are An AlphaMania 1.0 User*

If you are already an experienced AlphaMania 1 user, you will want to read “What’s new in AlphaMania 2” in this Part and then skip to Part 6.

## *What AlphaMania Does*

AlphaMania does for graphics what Director Rich Text does for text - it eliminates jaggies! Whether you are a Director developer, an artist, or both, you can now use Director in a whole new way. Using the same techniques you already know from your favorite graphics editors, you can create stunning, time-saving, never-before-possible transparency effects in Director 5 & 6. A few examples:

1. Jeff is a brilliant artist who isn’t much of a technologist. Give him a paint program and he’ll create anything you like, but tell him to work in Director and he’ll dig in like a mule. Why? Because in his paint program he can move elements around seamlessly. When he moves sprites around in Director, however, their nicely anti-aliased edges suddenly become ugly because they are only anti-aliased to one location on one background. This drives him crazy. The solution: AlphaMania. He can easily import his graphics into Director complete with anti-aliasing (alpha-channel) information from his image editor and then change their locations or backgrounds at will without ruining the sprite’s smooth edges.
2. Imagine an interactive kiosk for kids requiring that the user be able to switch out many elements of a static nature scene. They will be able to change the sky from blue to stormy or the field from brown to green. Scott, the artist and lingo programmer on the project, is dreading the number of elements that will be required to provide this functionality while preserving the high quality of the individual graphic elements. A foreground tree, for example, will have to be split into multiple sections dictated by the different parts of the image it is in front of. Each of these sections will require multiple versions: the top of the tree as anti-aliased to the blue sky, the top of the tree anti-aliased to the stormy sky, etc. Scott has calculated that what intuitively should be just one tree graphic will now require over fifty smaller graphics and a lot of intense programming to keep track of which version of each part of the tree corresponds to which background. The solution: AlphaMania. Just import the tree as a single AlphaMania cast member and it will automatically anti-alias to the changing backgrounds.
3. Damian needs to create a complex animation for frame-by-frame output to video. He’d like to use Director’s powerful animation tools and score metaphor, but he’ll be using a lot of transparency and shadow effects on elements of his animation that will be moving across the screen over varied backgrounds. Solution: AlphaMania. AlphaMania supports the same kinds of alpha-channel effects within Director that video editors have been using for years with products like Premier and After Effects. (If you are not sure what an alpha channel is, please turn to the Reference section and read the sub-section entitled “Anatomy of an alpha channel/” for a full discussion.) Not convinced yet? Take a look at the sample movie accompanying this download!

## AlphaMania Is Easy To Use

Xtras and XObjects are the tools with which programmers can extend Director's abilities. Unfortunately all XObjects and most Xtras require lingo programming to use. This can be time-consuming, difficult, and frustrating. AlphaMania, however, is a "Cast-Member Xtra." This means that you use it just like any other cast member type. You import an AlphaMania cast-member with a menu command and it appears in the cast with it's own thumbnail preview. You can drag it to the stage or the score. You can reposition it and in-between it just like a bitmap cast member, all without using lingo. Like other cast member types that Director supports internally, AlphaMania sprites and castmember have advanced properties that can be tested and set through lingo. If you don't need these advanced features, AlphaMania doesn't require a single line of lingo programming!

And best of all, AlphaMania is free to distribute with your multimedia productions. There are no royalty or distribution fees!

## What's new in AlphaMania 2?

AlphaMania 2 has many new capabilities that allow you to expand your authoring repertoire in exciting ways, so be sure to investigate them thoroughly. In this section we will provide a quick overview of the new capabilities and refer to other sections where they are discussed in detail.

### **Batch Importing**

The most popular new feature in AlphaMania 2 is the ability to import multiple files (or Photoshop layers in conjunction with PhotoCaster 1.2 or later) at a time. See "Batch Importing" for more details.

### **Drawing Methods**

You can now use AlphaMania castmembers as masks to reveal other castmembers! In this case the RGB data of the AlphaMania castmember is ignored and the alpha channel is used as a mask to reveal parts of a bitmap castmember that you specify. No Effector Set Xtra is required for this feature. See "Drawing Modes" for more details.

### **8-bit only AlphaMania castmembers**

In order to conserve memory in those cases when you only want to use an AlphaMania castmember as a mask, you can import only a graphic's alpha channel, ignoring its RGB data. See "8-bit only AlphaMania castmembers" for more details.

### **Effector Set Support**

Media Lab has introduced a line of Xtras, the Effector Set Xtras for AlphaMania. These xtras allow AlphaMania 2 to create stunning dynamic special effects in Director. See "Using Effects" for more details.

## Scaling

AlphaMania 2 now allows scaling larger or smaller than the original size of the graphic, with an option for interpolation for higher image quality. This is implemented as an Effect, exactly like the effects found in the Effector Set Xtras. You can control scaling from lingo or from the SetFX movie in the Xtras menu. See “Scale Effect” for more details.

## The SetFX Movie

This is a point-and-click interface for setting various attributes of your AlphaMania castmember. It is primarily useful for modifying the special effects provided with the Media Lab Effector Set Xtras, but is also used to manipulate the drawing method and the Scale effect, which pertain to AlphaMania 2.0. If you loathe and fear lingo, this is the tool for you. If you’re a lingo wizard, you will still want to examine this movie because it can write much of your effects-specific lingo code for you.

## New Lingo

New lingo has been added to support many of the new features. These include:

- GetPixelAlpha now works on the sprite level, even with effects like scale and rotate
- GetEffectList, DescribeEffect, and other useful effects-specific commands have been added (See “Using Effects with Lingo”)

## Changed Defaults for some settings

Some defaults for newly created castmembers and sprites have been changed by popular demand:

- Dither In 8-Bit now defaults to on
- the clickThreshold now defaults to 1

## Many Other Improvements

Including smaller RAM footprint, translucent thumbnails, faster thumbnail drawing, etc.

## *Migrating from AlphaMania 1 to AlphaMania 2*

AlphaMania 2 castmembers cannot be loaded or used with AlphaMania 1. AlphaMania versions 1 and 2 cannot be loaded simultaneously. When you open a movie containing AlphaMania 1 castmembers when AlphaMania 2 is installed, they are automatically upgraded if you are a registered user. If you save the movie you will no longer be able to use the AlphaMania 2 members with AlphaMania 1! If you are using an unregistered version of AlphaMania 2, you can try out the new features on version 1 castmembers, but they will not be saved as version 2 and will lose any AlphaMania 2 settings you may have applied.

## Part 2: Users Guide

If you are not sure what an alpha-channel is, please turn to the Reference section and read the subsection entitled “Anatomy of an alpha-channel” for a full discussion. You’ll be glad you did.

### Operating Systems Supported

#### **Mac OS**

AlphaMania for Mac OS supports system versions 7.0 and higher, and supports both the PowerPC and 68K platforms.

#### **Windows 95 / NT**

AlphaMania 2 for Win 32 runs in conjunction with Director 5 on the Windows 95 or Windows NT platforms. Windows 3.11 and earlier versions are not supported. Windows 3.11 may be supported in the future for playback only. When this occurs, all registered users of AlphaMania will receive notification.

### Installation

In order to install an Xtra, you simply place it into the “Xtras” folder inside your Director folder. Place the AlphaMania Xtra inside your Xtras folder and then run Director.

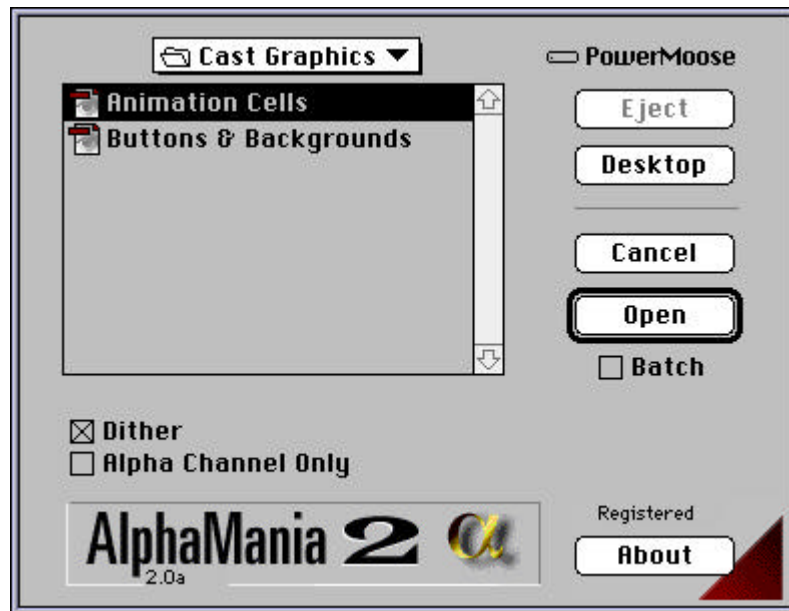
The command to create an AlphaMania cast member is found, like other cast member types, in the “Insert” menu. Choose Insert->Media Lab Media->AlphaMania. If this menu choice does not exist, AlphaMania is not installed correctly.

### Registering & Unlocking AlphaMania

Refer to Part 7 of this manual.

## Importing PICT and Targa Files

After selecting Insert->Media Lab Media->AlphaMania you will then see this dialog (the dialog will appear slightly different under Windows):



### **File Import Options:**

#### *Batch:*

Checking this box will import all Targa (and PICT under MacOS) files in the selected folder when you click the open button. (See “Batch Importing Multiple Files”)

#### *Dither:*

Turns the “dither in 8-bit” option on or off for imported AlphaMania castmembers.

#### *Alpha Channel Only*

If you wish to import only the alpha channel from a file for use as an 8-bit only AlphaMania castmember (See “8-bit Only AlphaMania Castmembers” ) then select this checkbox.

Both the Mac OS and Windows versions allow you to select a TGA file. (TGA files are also known as Targa files.) In addition the Mac OS version will allow you to select a Macintosh PICT file. With the addition of Media Lab’s PhotoCaster 1.1 Xtra (or later) you can also choose an Adobe Photoshop file from which to select layers.

NOTE: The Mac OS version respects DOS-style file name extensions. If a Targa or Photoshop file is brought to the Mac from a Windows machine and as a result does not have the proper Mac OS four-character file-type, AlphaMania will still recognize it if it has the proper three-letter name extension. AlphaMania recognizes the following extensions:



.pct	Mac OS PICT file
.tga	Targa file
.psd	Photoshop document (If PhotoCaster 1.1 or later is installed)

If you have chosen a PICT or Targa file, that's it! Click "Open" and your file will be imported. If you have chosen a Photoshop file, see "The AlphaMania-PhotoCaster Connection." Note: AlphaMania will only import 32-bit (true-color) files that contain alpha-channel data.

## ***Batch Importing***

AlphaMania 2 allows multi-file import and, with Media Lab's PhotoCaster Xtra (v1.2 or later), multi-Photoshop layer import.

### **Importing Multiple Files**

To import multiple files under follow these steps:

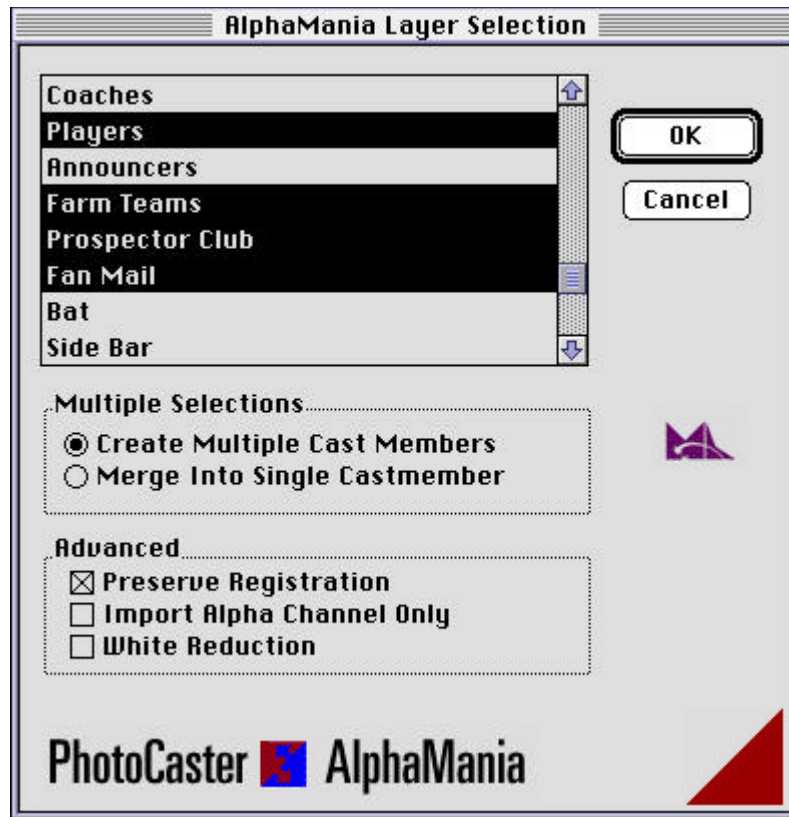
1. Collect all of the files you want to import into a single directory.
2. In Director, open the AlphaMania import dialog from the Insert->Media Lab Media menu.
3. Select the "Batch Import" checkbox.
4. Navigate to the inside of a directory of files you want to import and select any file.
5. When you click the "Open" button, all of the files in the directory will be imported into the cast in alphabetical order. All options set in the import dialog will be applied to all imported castmembers.

### **Importing Multiple Photoshop Layers**

PhotoCaster is another time-saving Xtra from Media Lab that allows you to import Photoshop layers as individual pre-positioned bitmap cast members in one easy step. (A fully functional Demo that even works with AlphaMania can be downloaded from [www.medialab.com](http://www.medialab.com))

If you have both AlphaMania and PhotoCaster 1.1 (or later) installed you can import Photoshop layers as AlphaMania cast members. (Note: PhotoCaster versions prior to 1.1 do not include this functionality. PhotoCaster 1.1 only allows import of one layer at a time. PhotoCaster 1.2 allows multi-layer import and is a free upgrade to registered PhotoCaster users.)

When you select a Photoshop file from the AlphaMania import dialog and click the Open button, you will be presented with the following dialog (the dialog will appear slightly different under Windows):



The list contains the names of the all the layers in the Photoshop document in order from the topmost to the bottom layer.

To import multiple Photoshop layers, follow these steps:

1. Select the layers you would like to import using standard multiple-selection techniques. (MacOS: Shift-clicking creates continuous selections, Command-clicking creates random selections. Windows: Shift-clicking creates continuous selections, Control-clicking creates random selections.)
2. When you click the “Okay” button all of the layers will be imported into your cast in alphabetical order.

### Layer Import Options:

#### *Multiple Selections*

If you choose multiple layers you may wish to import them as separate castmembers or you may wish to merge them into a single castmember. Use these radio buttons to choose.

### *Preserve Registration*

If this option is chosen, your AlphaMania cast member will have a registration point (see *reggaePoint* below) relative to the center of the stage that matches the imported layer's position relative to the Photoshop document. This means that if you drag the AlphaMania cast member to the score it will appear on the stage in the same place it appeared in the Photoshop document.

### *White Reduction*

This option is provided as a convenience to PhotoCaster users. It causes any pure white in the Photoshop layer / layers to be replaced with the same shade of off-white that PhotoCaster uses with its own White Reduction option. This allows AlphaMania cast members with white in them to match bitmap cast members imported with

PhotoCaster. See the PhotoCaster documentation for more information about White Reduction.

### *Import Alpha Channel Only*

If you wish to import only the alpha channel from a layer for use as an 8-bit only AlphaMania castmember (See "8-bit Only AlphaMania Castmembers" ) then select this checkbox.

### **Batch Import Special Considerations**

Just because you *can* do something doesn't mean you *should*! Now that batch importing has been implemented it will be very tempting to import huge numbers of graphics as AlphaMania castmembers. Please remember that the flexibility and power of AlphaMania castmembers comes at the expense of higher memory usage. We recommend that background graphics and other elements that don't require this flexibility still be imported as normal bitmaps.

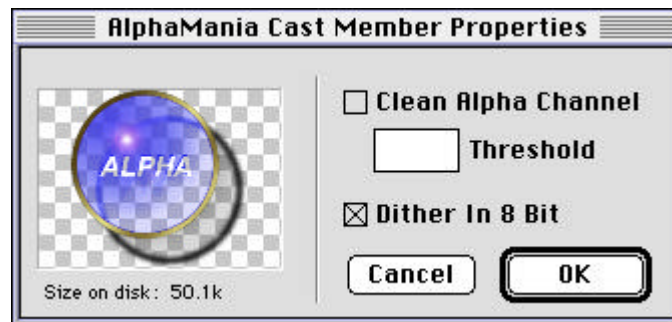
Batch importing can be a very memory-intensive task and you may not be able to import a large folder of files. It is best to divide the batches up into smaller directories and save after each multi-import. The number of files/layers you can import at once is dependent on the following factors:

1. Amount of free memory. Your memory can be used most efficiently during import if you save the Director movie and cast files before importing. Also, be sure to turn "Use temporary system memory" on in the General Preferences dialog before importing, and quit other active applications if necessary.
2. Size of files. This means dimensions, not size on disk. Some files can be very small on disk but take up a lot of RAM once they are imported. The RAM requirements of an AlphaMania castmember are directly proportional to that member's horizontal and vertical dimensions.

## AlphaMania Basic Cast Properties

AlphaMania, like other Director cast member types, supports a variety of cast member properties. Some are editable via the Cast Properties dialog, some by lingo commands, and some by both methods. We'll cover the Cast Properties dialog first.

There are two ways to bring up the AlphaMania cast properties dialog. The first is by double-clicking the cast member. The second is by clicking the "options" button in the Cast Info dialog. Either method will present the following dialog (the dialog will appear slightly different under Windows):



### **The Cast Properties Dialog:**

The cast properties dialog presents the following properties:

#### *Clean Alpha Channel*

This property can speed up your AlphaMania castmember. It allows you to set a cutoff threshold that will force alpha values that are close to transparent to become fully transparent and will force alpha values that are close to opaque to become fully opaque. This allows the drawing engine to be more efficient. The range is 0-127. You can experiment with different values to achieve the best results. Larger values can seriously degrade image quality, depending on the image.

#### *Dither In 8 Bit*

This option improves image quality at a small speed cost. It uses a dither filter on opaque areas of the AlphaMania cast sprite to improve the 8 bit representation. Experiment to see what works best with a given graphic.

The Cast Member Properties dialog also shows a thumbnail preview of the graphic and how much space the cast member occupies on disk. AlphaMania uses a proprietary compression scheme to minimize disk and memory use. See "Disk Space Issues" in the "Optimizing AlphaMania section below.

### **Basic Lingo Cast Properties:**

This section ignores properties involving Drawing Methods or Special Effects. See those sections for more info on AlphaMania 2.0 properties.

#### *ReggaePoint*

No, it's not a joke. This property corresponds to the regPoint property of bitmap castmembers. It takes a point as an argument and determines where the sprite is drawn relative to its hLoc and vLoc.

The default reggaePoint is centered horizontally and vertically in the cast member unless you have imported from a Photoshop layer with the "Preserve RegPoint" option. (See "Importing Photoshop Layers" above.)

You can test and set this property via lingo. The coordinates are relative to the upper left corner of the castmember, not the stage. For example:

```
-- The following sets the reggaePoint of the cast member named "AlphaManiac"
-- to its upper left corner.

set the reggaePoint of member "AlphaManiac" to point (0,0)

-- The following sets the reggaePoint of the cast member named "AlphaManiac"
-- to its lower right corner.

set rightEdge = the width of member "AlphaManiac"
set bottomEdge = the height of member "AlphaManiac"
set the reggaePoint of member "AlphaManiac" to point(rightEdge, bottomEdge)
```

For more information about cast-member registration points, consult your Director documentation.

NOTE: We would have named this property "regPoint," but Director intercepts and rejects that property name before an Xtra can detect it. If Macromedia changes this in the future, you will be able to use the word regPoint with the current version of AlphaMania and a future version of Director. For now ReggaePoint will have to do...

#### *BoundsRect*

This property corresponds to the rect cast property of bitmap castmembers. It is a rect value, and can be gotten, but not set.

#### *Dither*

This property is true or false and determines matches the Dither in 8 Bit cast property described above. It can be gotten or set.

```
set the dither of member "AlphaManiac" to true
```

## AlphaMania Basic Sprite Properties

NOTE: Until Macromedia implements a way for Xtras to present sprite properties dialogs and save sprite properties in the score, all Xtra sprite properties must be set via lingo.

### **clickThreshold**

This property controls how an AlphaMania sprite determines whether or not a mouse click has occurred within it. It determines this by checking the alpha-value for the pixel that was clicked on. If this value is greater than or equal to the clickThreshold, the mouseDown is passed on to lingo. (If you aren't sure what we mean by "the alpha-value" see the section "Anatomy of an Alpha Channel" below)

This property can be tested and set. For example:

```
-- The following command will cause the AlphaMania sprite in channel 1
-- to respond to clicks anywhere within its sprite rectangle

set the clickThreshold of sprite 1 to 0 -- that's a zero, by the way

-- The following command will cause the AlphaMania sprite in channel 1
-- to respond to clicks on pixels within its sprite rectangle that have
-- alpha values greater than or equal to 50

set the clickThreshold of sprite 1 to 50
```

This property allows arbitrary click detection areas, something never before possible in Director. For example, you can make donut-shaped sprites that don't respond to clicks in the hole -- perfect for that "Dunkin' Donuts Interactive" CDROM.

NOTE: Sprite properties are not saved with the movie and are not retained by the sprite when the movie is stopped. See the Director documentation for additional information about sprite properties.

### *AlphaOnly*

Use this property if you want to know whether an AlphaMania castmember is only an alpha channel with no RGB data. It returns true if the member is 8-bit only. Example:

```
if the alphaOnly of member 1 = true then beep
```

### **AlphaMania Custom Functions**

AlphaMania several custom functions that you can use with AlphaMania sprites. We will describe GetPixelAlpha here. The remaining custom functions are used with effects and are described in Part 6.

### *GetPixelAlpha*

Use this function to determine the current alpha value of any pixel within an AlphaMania sprite. It takes a sprite and a point as arguments. The point is relative to the stage. This routine takes into account any effects that have been applied. Returns zero if the point is outside of the area of the sprite. Example that beeps if the cursor is over a non-zero pixel of our sprite:

```
if GetPixelAlpha(sprite 3, point(the mouseH, the mouseV) > 0 then
    beep
end if
```

## *Distributing AlphaMania 2 with Projectors & Shockwave*

There is no license fee for distributing AlphaMania with your projectors or Shockwave movies. You can even let users download it from your site or link directly to Media Lab.

### **Projectors**

Distributing AlphaMania 2 with projectors is as simple as ensuring that there is a copy of AlphaMania in a folder named “Xtras” that is in the same folder as your projector. If you want to distribute a projector on a different platform from the one you authored with you just have to download the appropriate demo from Media Lab and place it in the appropriate Xtras folder.

Director 6 makes this even more seamless because it allows you to package Xtras into your projectors. AlphaMania will automatically be included in projectors you create. If you are using any effects other than Scale in your movie you will need to add the appropriate Effector Set explicitly when you create the projector. Add it in the same way you add movies in the Create Projector dialog.

### **Shockwave**

Unfortunately Macromedia has yet to provide a seamless way for Xtras to automatically download when they are needed by Shockwave movies. If you think this feature is important, let them know!

Until they develop a better way, anyone viewing a Shockwave movie with an AlphaMania sprite must have AlphaMania (and any required Effector Set Xtras) installed in the Xtras folder that Shockwave creates. Under the MacOS this folder is in the same folder Shockwave is in, for example the Netscape Plug-Ins folder. Under Windows with Internet Explorer this folder resides in the windows\system\macromed\director\ directory.

## Part 3: Reference

### *Anatomy of an Alpha Channel*

Computer images are made up of individual dots called pixels. Each pixel can be thought of as a single square on a piece of graph paper. Each pixel has to have some method for determining what color it is. There are many ways to tell the computer what color to show a pixel as, and all involve assigning numbers to different color values. For example you can specify the color of a pixel by giving a number each to its hue, saturation, and brightness. The most popular way to describe a pixel's color is known as RGB, or Red, Green, and Blue. You can create most visible colors by mixing red, green, and blue in different proportions. Every pixel in an RGB image has three numbers that are associated with it that represent that pixel's red, green, and blue proportions. These numbers, for various reasons, fall in the range of zero through 255. A pixel that has the maximum red value (255) and the minimum green and blue values (zero) is red. The red values for every pixel in a given image together make up what is known as the red channel. The same goes for green and blue.

An RGB image that is 32 pixels wide and 32 pixels high has 1024 pixels. ( $32 \times 32 = 1024$ ) Since each pixel has three color values (red, green, and blue) associated with it, there are actually  $1024 \times 3$  numbers that completely describe that image, three numbers for each pixel.

This system works well within a single graphic, but more information about each pixel is needed if you want a graphic to be semi-transparent so that it reveals graphics that are behind it. (We might be talking about layers in Photoshop, tracks in Premier, or sprites-channels in Director to name a few instances of layered graphics.) Graphics programmers needed a way to tell if a given pixel is totally transparent, partially transparent, or totally opaque. This is where the alpha channel comes in. Another number in addition to the red, green, and blue values was added to each pixel to represent that pixel's transparency. The alpha values for every pixel in a given image together make up what is known as the alpha channel. This means that images which have alpha channels are larger. If we return to the  $32 \times 32$  image mentioned above, we see that the additional channel means that we now need  $1024 \times 4$  numbers to represent the image. The image is larger but it is now much more useful.

For example, we can represent a transparent drop-shadow as an area of solid black with its RGB values (black means R, G, and B would all be zero) but with a gradient in the alpha channel. We can represent a glow with an area of solid white in the RGB values (white means R, G, and B would all be 255) but with a gradient in the alpha channel. (See "Alpha Channel Techniques" below) A pixel with an alpha value of zero is totally transparent and invisible, therefore its RGB values are irrelevant. By the same token a pixel with an alpha value of 255 is totally opaque, and we can ignore any pixels that are behind it.

Looked at from a more right-brained point of view, using an alpha-channel can be compared to painting on glass. If you paint on glass with a thick oil paint you won't see anything through the



glass. This is like using an opaque alpha channel. But if you paint with a watery powdered tempera, the paint will be translucent. The more water you add to the paint, the more translucent it is, just as the lower a pixel's alpha channel value the more translucent it is.

Of course the term "translucency" is a bit of a misnomer because the computer screen doesn't really have "layers." Translucency is just another way of talking about color mixing. If our top layer is red and 50% translucent and our bottom layer is green and opaque, then the color of the topmost pixel will be reddish-green. Change the background to blue and the final color will change to reddish blue. The best way to fully understand these concepts is to see an example of alpha channels in action.

Examine Figure 1. In it we represent alpha channel values as a grayscale graphic that is the same size and shape as the source RGB graphic. Black pixels in the alpha channel represent maximum opacity (255), white pixels represent maximum translucency (or zero, which means the corresponding RGB pixel will not be mixed with the background), and gray pixels represent intermediate values by how close to white or black they are. We start on the left of figure 1 with two red squares as defined by their RGB channels. We then add an alpha channel to each square that changes the appearance of the square. In the smaller example it is easy to see that no mixing between red and green occurs in the middle of the square because the middle of the alpha-channel is black and therefore signals that the middle of the RGB data is to be copied without mixing. At the edges of the alpha channel we see values that are neither completely black (opaque) nor completely white (transparent.) In the resulting graphic over the green background we see that those pixels caused a mixing of the red and green, giving red a higher concentration where the alpha channel is darker (the middles of the edges) and green a greater concentration where the alpha channel is lighter (the corners.) If we changed the background from green to blue we would see similar behavior, the red would mixing with blue wherever there were intermediate alpha values.

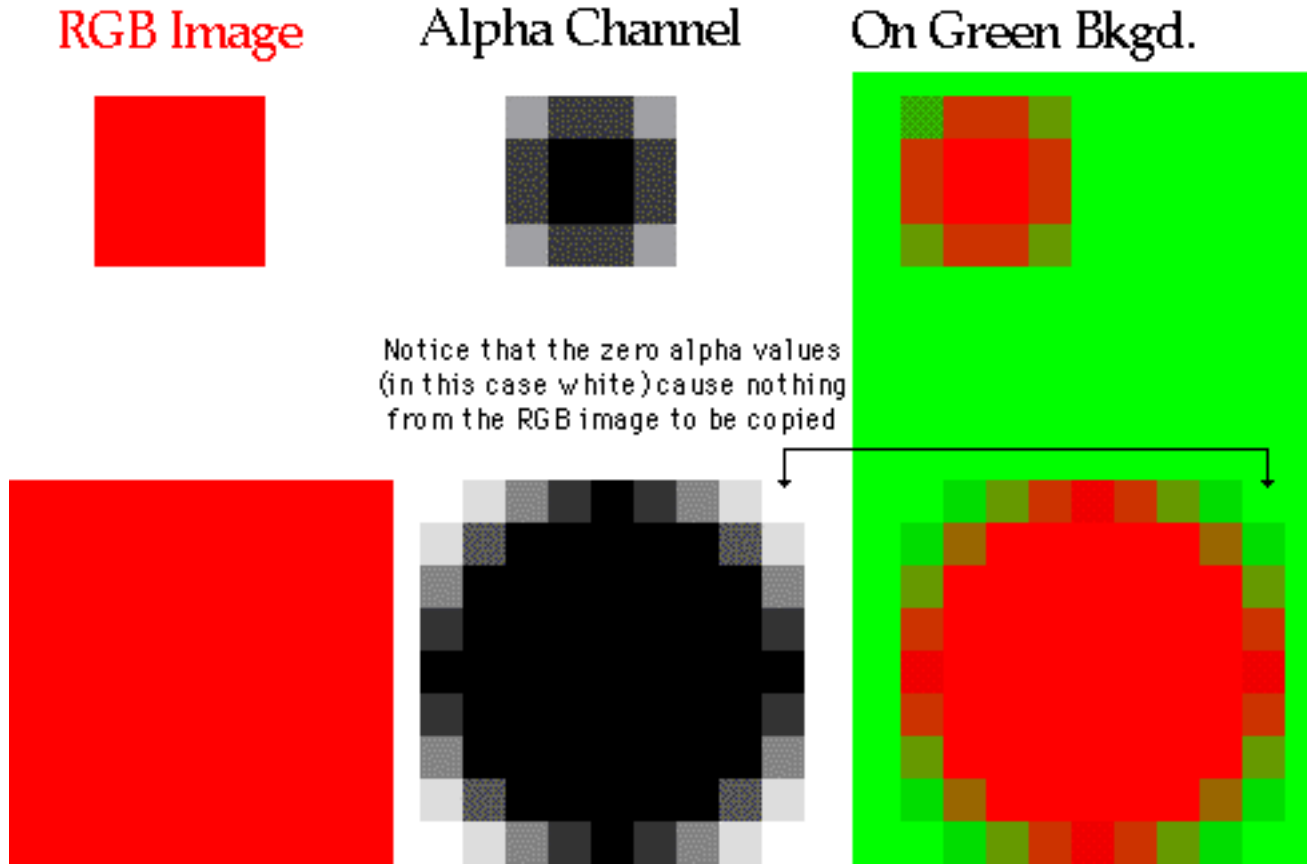


Figure 1.

Although it is important that you understand how and why alpha-channels work, luckily you almost never have to manipulate these numbers directly. Paint programs, for example, allow you to choose the RGB values of your paint-brush by simply clicking on a palette of colors. Then, by painting with the brush you are actually modifying the RGB values of the pixels you paint over. Editing the alpha channel values can also happen transparently. (pun intended) In Photoshop 3.0 or xRes, for example, you can choose a brush that is only 50% opaque. When you paint with it, background layers are still partially visible through whatever color you are painting with. This is because the brush is creating 50% alpha channel values for the same pixels that it is adding RGB values to. (A fifty percent alpha value is halfway between 0 and 255, or 127.) Every layer in a Photoshop 3.0 document has an alpha channel that keeps the transparency values for every pixel in that layer. What looks like a feathered shadow to you looks like a gradient in a layer's alpha-channel to the computer.

### *Creating and Editing Alpha Channels*

There are two ways to create and edit alpha channels: directly, and indirectly. When editing a layer in Photoshop, every time you use a tool with transparency or anti-aliasing you are indirectly modifying an alpha channel for that layer. We say "indirectly" because you don't explicitly create the alpha channel and you cannot view or edit it separate from its RGB channels. But it is always there, and

it always affects the relationship between the layer and any layers behind it. In addition to this implicit alpha channel, you can add what is known as a “layer mask.” This is basically an additional alpha channel that works like the implicit channel, but can be viewed and edited by itself as a grayscale image. (When PhotoCaster imports Photoshop layers into AlphaMania cast members, it supports both the implicit alpha channel and any layer mask.) Another way to directly create and edit an alpha channel in Photoshop is in a single layer document. You can create a document with one layer and add a channel to it. This channel will be interpreted as that document’s alpha channel if you save the document as a PICT or a tga file. Single layer Photoshop files don’t have the “implicit” alpha channel described above.

Note: One confusing thing occurs when directly editing an Alpha channel is that sometimes white represents opaque (255) and black represents transparent (zero) and sometimes it is the reverse. Neither convention is any better or worse, but you need to know which system you are dealing with. Some applications, like Photoshop, allow you to specify which convention to use.

Sample files included with the downloadable version of AlphaMania demonstrate all of these techniques. See “Alpha Channel Techniques” below for more information.

3D renderers are programs that fall into the “implicit” alpha-channel category because they go about the work of creating alpha-channels automatically. In order to edit the alpha channels of 3D rendered graphics, you must open them into a 2D editor. Some 3D programs only use alpha channels to provide anti-aliasing effects. (See “Anti-Aliasing” below) Others, notably Electric Image, can use alpha channels to provide a wide range of transparency effects. In 3D software you want to render to an alpha channel straight - pre-multiplying will render the channel and create a fringe (EIAS).

## *Techniques for Creating Alpha Channel Effects*

### **Gradient Transparency Title Backgrounds**

A simple but dramatic effect currently enjoying great popularity among creators of titles and other screen data for big-budget television network news and sports shows is varied transparency title backgrounds. We’ll create an example. Let’s say we’d like to have a blue horizontal bar that is opaque at its left extreme and gets progressively more transparent as it goes to the right. We could use this as a subtle background to put titles over which would still allow the viewer to see what’s behind the blue bar. If you have read and understood the anatomy of alpha-channels you can probably already guess what you need to do to create this

A good strategy for deciding how to create a particular effect is to divide the elements you’ll need into two categories: the RGB graphic and the alpha channel. For the above effect we know we need two things:

RGB: A solid horizontal blue bar

Alpha: A grayscale gradient from black on the left to white on the right.

We'll create this effect with a one layer Photoshop document. The same technique would work in xRes, etc.

Steps:

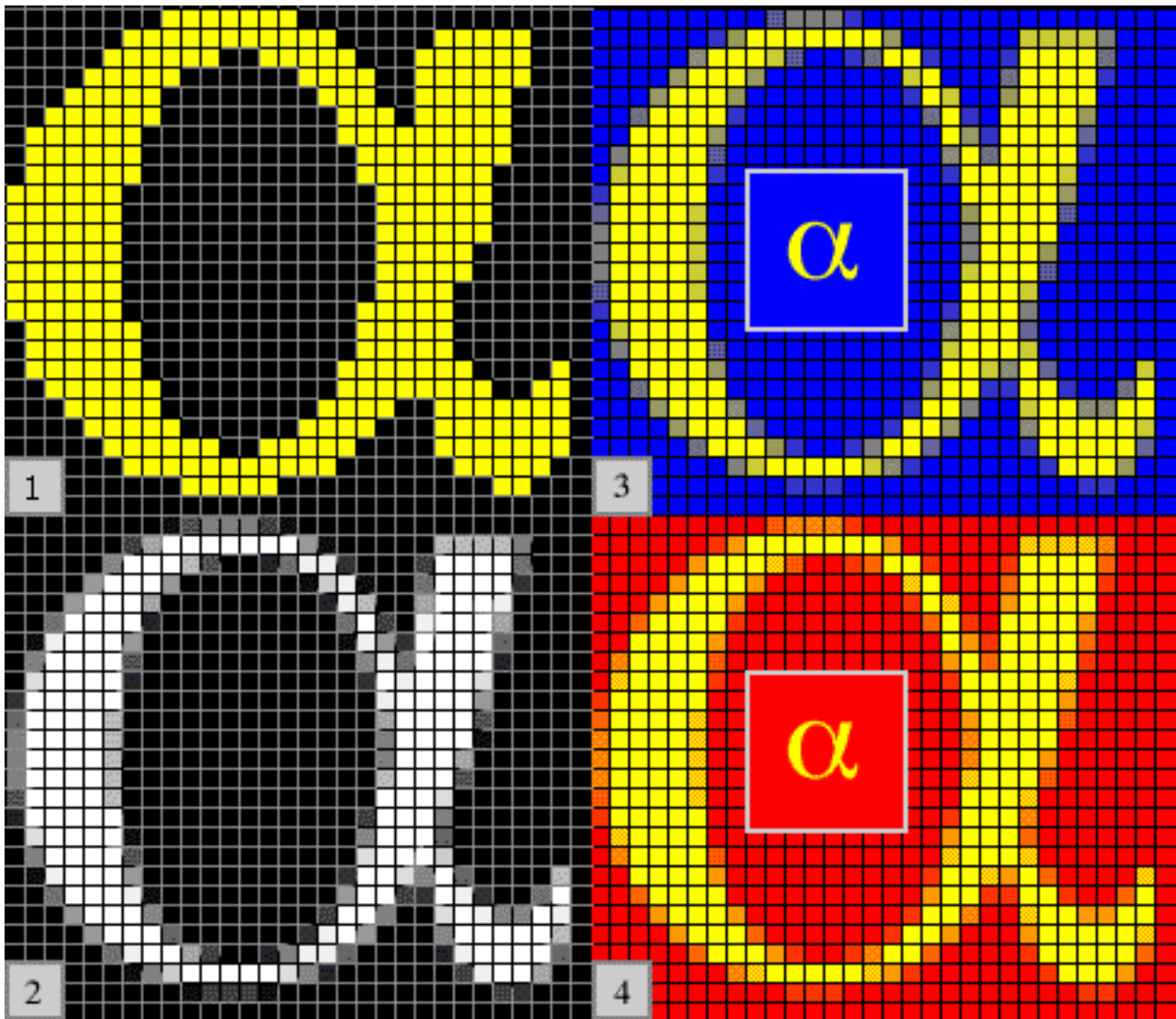
1. Create a new Photoshop document that has the dimensions you want the horizontal bar to have. Let's say 50 tall by 200 wide.
2. Fill the entire document with solid blue (not a gradient).
3. Add a channel in the "channels" palette and choose it for editing. Hide the RGB channels so they don't distract you.
4. Use the gradient tool to create a gradient of white to black from right to left
5. View all channels. You will see an approximation of the transparency with the preview color assigned to the alpha channel, usually red. The red color shows through in proportion to how much of the background you will see when using this graphic in Director.
6. Save as a Targa file and import into Director with AlphaMania
7. Drag it the stage. Neat, huh? Now change the color of the stage or put the banner on top of other sprites and watch how the semi-transparent parts of the banner automatically adjust.

Here is what this will look like:



### Anti-Aliasing

Anti-aliasing is also achieved with alpha channels. Anti-aliasing is a way of blurring and therefore smoothing the harsh, stair-stepped line that can occur on the edge between areas of different colors on a computer screen. This effect is created by generating pixels along this edge that are intermediate colors between the two extremes. If you look at a magnification of an anti-aliased edge you can see these pixels.



**Figure 2.**

Examine Figure 2. The  $\alpha$  symbols is in the top layer, layer 1. The solid red and blue areas are in the background layer. (Grid lines have been added for easier comparison of pixels.) Corner 1 represents the RGB data. Notice that its edges are completely solid. Corner 2 represents the alpha channel for corner 1. In this case white represents opaque and black represents transparent. Corner 3 shows what happens when the RGB data of corner 1 is drawn over blue using the alpha data shown in

corner 2. The inset shows the beautifully anti-aliased character unmagnified. Corner 4 shows what happens when the RGB data of corner 1 is drawn over red using the alpha data shown in corner 2. The RGB and alpha data of layer 1 never change but yet the combination allows the graphic to be drawn over any color area and remain anti-aliased. (Of course this works equally well with graphics that aren't solid colors, it's just harder to see what's going on.)

Do you think corners 3 and 4 would appear differently if corner one were filled with a solid square of yellow instead of the alpha symbol?

The answer is no. Corners 3 & 4 would be drawn exactly the same if corner 1 was a solid square because the black areas of corner 2, the alpha channel, are already causing the RGB pixels that correspond to them to be ignored (masked). Only pixels with non-zero alpha data will be drawn. This is very important to understand.

NOTE: Figure 2 has been dithered so that those of you viewing this document in 8 bit can still see the subtleties of the color mixing.

### Drop Shadows

Drop shadows that darken but don't obscure what they are on top of are easy to achieve with AlphaMania. Examine Figure 3. It is a 4:1 magnification. The gray grid pattern is the background and useful for showing which pixels are transparent. This image is included in the AlphaMania download in Targa and PICT formats so you can examine it in your favorite editor and try it out in director.

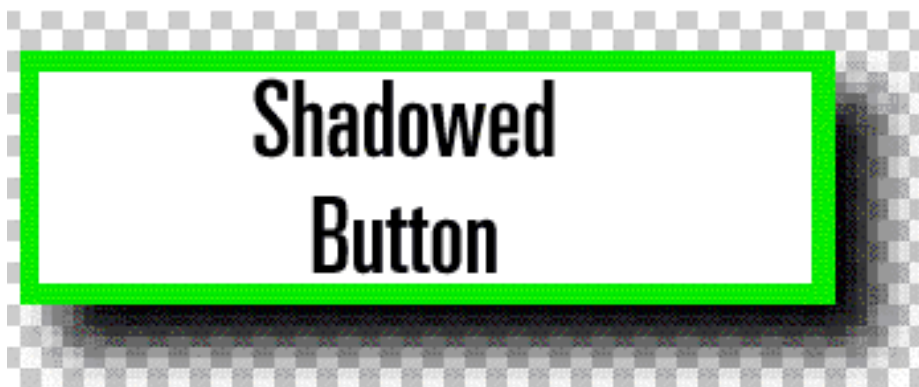
Use nice text here...



Here we see the RGB data. Because we want the color of our shadow to be black, we place a solid black area as large as the area we'll want our shadow to have below and to the right of the image.



This is our alpha channel. White is opaque. Notice the area corresponding to the area of the button itself is solid white. That area is to be entirely opaque. The area of the shadow contains a gradient from opaque to transparent.



Here is the final result in Photoshop after we apply the alpha channel as a layer mask to the layer with the RGB data.

**Figure 3.**

Like most aspects of multimedia, alpha channels seem difficult at first, but the more you use them the more comfortable and useful you will find them to be! Check Media Lab's web site for additional tutorial examples as we make them.



## Part 4: Optimizing AlphaMania

This section will give you all the information you need to get the most out of AlphaMania.

### Speed Issues

#### **Color Depth**

Because the colors needed by AlphaMania are constantly changing to match a changing background, 16 and 32 bit screen depths are fastest. When AlphaMania is running in eight-bit it is constantly having to search for colors for every transparent pixel, and this can slow it down.

#### **Cropping**

AlphaMania sprites don't mind being cropped off of the top and bottom of the screen, but a significant slowdown can occur if they are cropped off of the right or left sides. Keep this in mind when animating AlphaMania sprites.

#### **Clean Alpha Channels**

AlphaMania doesn't perform the calculations necessary to achieve semi-transparency for opaque or completely transparent pixels. Keep this in mind when preparing graphics for import. Slightly opaque or slightly transparent pixels leftover from editing can really slow down the performance. This "junk-data" can be difficult to see while editing a graphic, so we recommend that you select all of a documents seemingly empty space and delete it just before importing into Director. A sure way to tell if there is junk data is if AlphaMania does not seem to crop your graphic to the rectangle containing only the visible pixels.

You can use the Trim Alpha Channel cast property to make AlphaMania skip this "junk-data".

#### **Blend Ink Effect**

While AlphaMania supports the blend ink effect, it can slow down drawing.

#### **Layered Sprites**

Like other Director types the following rules apply: AlphaMania sprites draw more slowly the more sprites they are on top of. (This is especially true of drawing over other AlphaMania sprites.)

AlphaMania sprites draw faster over bitmaps than over object-oriented graphics.

### Space Issues

#### **RAM**

As always, the smaller your cast member the better. The amount of RAM used by an AlphaMania cast member is directly proportional to the size of that member's rectangle. Because of this, when AlphaMania imports a graphic it automatically crops off unused white space at the edges. Make sure there are no elements which are drastically increasing the size of the rectangle.

### **Disk Space / Bandwidth**

AlphaMania uses a proprietary compression scheme to minimize disk and bandwidth requirements. In spite of the fact that 24 bit Director bitmaps have 33% less data than AlphaMania cast members, they are often the same size or larger on disk.

One of the compression method used by AlphaMania is called run-length encoding. This means that areas of similar color tend to compress more efficiently than areas of varied color. One way to shrink the disk size of AlphaMania cast members is to maximize areas of identical color. A solid red area will be much smaller on disk than a patterned red area.

To see how well your graphics are compressing, check the AlphaMania cast member properties dialog to see how much room they take up on disk.

## *Quality Issues*

### **Color Depth**

AlphaMania of course looks a lot better in 16 or 24 bit screen depths. Like other graphics, if you are using AlphaMania cast members on an 8-bit stage it is very important to provide an appropriate palette. One difference from normal 8-bit bitmap palette issues is that AlphaMania cast member will need colors available in the palette that approximate the colors created by mixing semi-transparent AlphaMania pixels with those below the AlphaMania sprite.

AlphaMania cast member will dynamically remap to whatever the current palette is. However if the palette changes while an AlphaMania sprite is on screen, you must move the AlphaMania cast member to a new channel so that it will be forced to remap. (This is a Macromedia problem that they are aware of.)

## *Limitations and known problems*

### **Re-import Crash**

**IMPORTANT:** There is a bug in Director 5.0 (and 5.01 but not 6.0 or later) which can cause crashes! Do not import an AlphaMania cast member into the same cast member slot from which an AlphaMania cast member was just deleted. Either import into a different slot or perform some undoable action between the deletion and the import. We are very sorry about this, but there is no way we can fix this. Macromedia is aware of the bug.

### **Trails**

AlphaMania trails behave a little differently than normal trails. First, they only work under MacOS at the time of this writing. Second, they draw on top of ANYTHING. Windows that are above the stage will be obliterated by trails. Even the menubar is not immune. Also, trails behave unpredictably on multiple monitor systems. We apologize for these problems, but felt that trails were too powerful to leave out until these problems were solved. We are working to improve trails and to implement them under Windows, and any progress we make will be included in a free

upgrade. Now the good news. AlphaMania trails are AWESOME! Experiment with them and see...

**32 Bit Files Only**

AlphaMania will only import true-color files that contain at least one pixel with non-100% transparent alpha data.

**Alpha Channel Required**

AlphaMania will not import files which have no alpha channel data. This includes files which have alpha channels that contain all zero values.

## Part 5: Frequently Asked Questions:

### **How does distribution work?**

You distribute the same Xtra that you have been using to author. Because it won't be used for authoring by your users, they will never have to deal with the copy protection issues, which only affect authoring. If you are authoring on a Mac, for example, but are delivering cross-platform you will need to download the Windows version of the AlphaMania demo and distribute it with your Windows project. There is no fee for the right to distribute AlphaMania with your product.

### **Does AlphaMania support linked cast members?**

No. The current architecture for Director Xtras (MOA) makes this difficult or impossible to implement. We will try to support this in the future.

### **Can we assign a palette to an AlphaMania cast member?**

No. AlphaMania uses the best colors available in the current palette for 8-bit playback. Choosing this palette carefully can drastically improve how AlphaMania cast members look in 8-bit.

### **Why are blue lines appearing in AlphaMania cast members that I import from PhotoCaster?**

The blue lines appear because you are using an unregistered copy of PhotoCaster. Once you purchase and register PhotoCaster, you will be able to import Photoshop layers without getting blue lines. PhotoCaster also serializes itself to a single copy of Director. (See below)

### **Why are red lines appearing in my AlphaMania castmembers?**

You are using an unregistered version of AlphaMania. Once you purchase and register AlphaMania, it will stop importing with red lines. Unregistered AlphaMania puts in red lines only at import. You can distribute your projects with AlphaMania, and not worry about your customers suddenly seeing red lines. Remember, AlphaMania serializes itself to ONE COPY OF DIRECTOR. If you register AlphaMania and try to use it with a copy of Director that has a different serial number, the blue lines will appear on import.

### **Hey, I moved my registered copy of AlphaMania to this other machine and I'm getting red lines when I import and an Unregistered message. What gives?**

When you register AlphaMania it locks itself to the serial # of the copy of Director you are running it under. Running it with a different copy of Director reverts it to demo mode. You can always re-register it if you have to move machines, re-install Director, etc. but it must be running with the same Director serial # that it was originally registered for. Simply re-enter your unlock code.

### **Can I give AlphaMania to a friend?**

Yes, please do. Just don't give them your unlock code.

### **How do I import Photoshop layers as AlphaMania castmembers?**

You need PhotoCaster v1.2, another Media Lab product. PhotoCaster specializes at importing Photoshop files into Director. AlphaMania will use PhotoCaster to provide this service.

**Where can I get PhotoCaster?**

The same place you got AlphaMania - from Media Lab! Check out [www.medialab.com/software](http://www.medialab.com/software), or see the section on contacting Media Lab, below.

**Hey, I've got PhotoCaster, but Photoshop files don't show up in the import dialog when I'm using AlphaMania. What should I do?**

You may have an older version of PhotoCaster. Be sure you are using version 1.1 or later. Also if you have brought Photoshop files from a Windows machine, make sure the file name ends with .psd. Example: WINPSFIL.PSD

## Part 6: AlphaMania 2 and Special Effects

This section discusses the many new features of AlphaMania 2 that facilitate the use of special effects and drawing methods. Subjects are broken down into techniques that don't require lingo programming and techniques that do. Lingo programmers should read all sections, non-programmers can skip the lingo-oriented sections.

(While the only effect included with AlphaMania 2 is scale, it is still important for anyone who wishes to get full use out of scale and the drawing methods should familiarize themselves with this section.)

### *Drawing Methods*

#### **Introduction**

An AlphaMania castmember contains RGB color information and transparency (alpha channel) information. (See "Anatomy of an Alpha Channel" if you are unsure about how AlphaMania's transparency works.) Normally the color and transparency information are used together to draw an AlphaMania sprite on the stage. If you import an anti-aliased red circle from Photoshop and drag it to the stage you will see an anti-aliased red circle. This is known as the "normal" drawing method.

In AlphaMania 2.0 we have expanded your ability to use AlphaMania castmembers by adding two other drawing methods, "reveal" and "apply". These two methods ignore the color information in your AlphaMania castmember, and use only the member's alpha channel. The color information for each pixel comes from some other source.

In the case of the "reveal" drawing method the color information comes from a bitmap castmember that you specify..

In the case of the "apply" drawing method, the color information comes from whatever is underneath the AlphaMania sprite that is in "apply" mode.

The below image shows all three methods in action. The Apply method is shown in conjunction with the HSB effect to achieve the yellowing of a background area. Notice how all three modes draw using the alpha channel of the AlphaMania castmember. The source bitmap shown at the top of the figure shows the bitmap castmember the reveal method is using at the bottom.

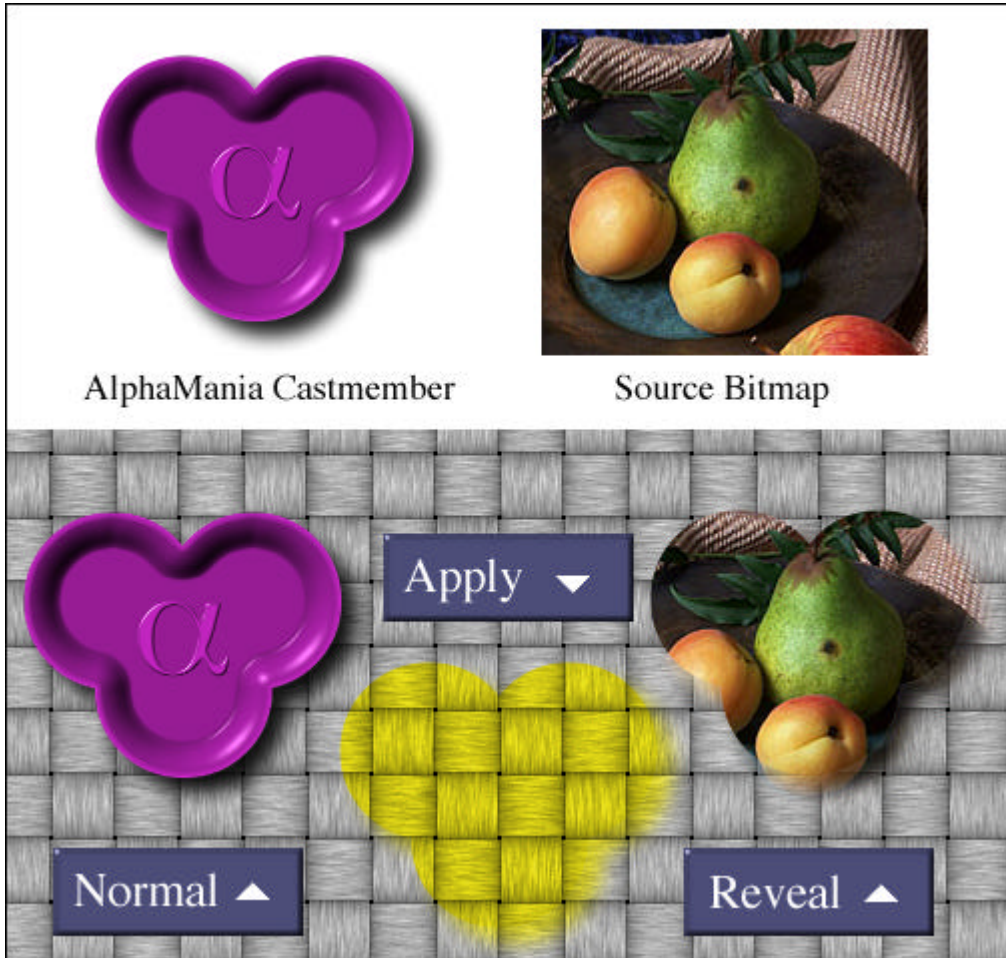


Figure 4. AlphaMania 2 drawing methods.

### Reveal Drawing Method Concepts

The reveal method allows you to “cut a hole in the stage” in the shape of the alpha channel of your AlphaMania castmember. Through this hole is revealed part or all of a bitmap castmember that you specify. It is important to understand that reveal does not cut through one sprite to reveal another sprite in a lower channel. What it reveals is another castmember, not another sprite. It is irrelevant whether or not the revealed castmember is also on the stage. By varying the amount of transparency of pixels in the AlphaMania sprite, you can blend the revealed graphic with the stage for dramatic effects.

#### *The SourceMember*

You can specify what bitmap to reveal through an AlphaMania sprite using either lingo or the SetFX movie. This property is known as the sourceMember. It is always a castmember property, so all

sprites created from a single AlphaMania castmember will reveal the same bitmap when the castmember is set to the reveal drawing method.

### *The SourcePosition*

Usually the bitmaps that you reveal using the reveal drawing method are not exactly the same size and shape as the AlphaMania castmember. If the sourceMember is smaller than the AlphaMania castmember, you may want to control where it appears within the AlphaMania sprite's rectangle on stage. If the sourceMember is larger than the AlphaMania castmember you may want to control which portion of the larger image is revealed through the smaller AlphaMania sprite. Both of these tasks are accomplished by adjusting the sourcePosition property of the AlphaMania castmember using either the SetFX movie or lingo. The sourcePosition property is a point. If you don't specify a sourcePosition, the sourceMember will automatically be centered. It is important to note that it is possible to set the sourcePosition to a point that shifts it out of the visible area of the sprite.

If you don't explicitly set the sourcePosition, AlphaMania automatically defaults to auto-center the sourceMember within the sprite or the stage, depending on which the source position is relative to. (See below.) If you set a sourcePosition and then later would like the source bitmap to auto-center again, set the sourcePosition to point(65535, 65535).

### *The SourcePosition Relative to the Sprite or the Stage*

Sometimes you will want different parts of the sourceMember revealed as the AlphaMania sprite moves around on the screen. For example you might have a large image of a human body on the stage along with a small AlphaMania sprite that reveals the skeleton of the body as the user drags it around. In this case the position of the revealed skeleton is constant relative to the stage, and doesn't change as the sprite moves. At other times you may wish the area of the revealed bitmap shown by the sprite to remain the same, even as the sprite moves around. In this case the position is relative to the sprite. The property that determines this is known as the sourceRelToSprite and is either true or false. It can be set with lingo or the SetFX movie. This property is also set at the castmember level and applies to all sprites created from the AlphaMania member. For an example of the use of this property, see the sample movie "basics.dir".

If you don't explicitly set this property, AlphaMania defaults to true, so that the source moves as the sprite does.

### *Reveal Drawing Method Considerations*

If you have specified the reveal drawing method but have not specified a sourceMember or if AlphaMania cannot find the sourceMember, the sprite will appear as a grayscale mask until you either specify a valid sourceMember or change the effect. It is important to note that it is possible to set the sourcePosition to a point that shifts it out of the visible area of the sprite, making the sprite itself invisible.

There is no "background transparent" mode for the source bitmap. If the source bitmap is against white, then the white, too, will be revealed through the AlphaMania sprite.



Because reveal uses a normal bitmap as its source, you can use this method to manipulate any normal bitmap as if it were an AlphaMania castmember. You can make a feathered mask that will anti-alias a standard bitmap, for example. You can also use a mask to rotate a standard bitmap.

### **Apply Drawing Method Concepts**

The apply drawing method uses an AlphaMania sprite's alpha channel as a mask to modify whatever area the sprite is covering on the stage. Exactly what modification occurs depends upon what effects have been added to the AlphaMania sprite or member. The amount of effect applied to each pixel on the stage is proportional to the opacity of the corresponding pixel in the alpha channel of your AlphaMania sprite, like adjustment layers in Photoshop 4. You could, for example, have an AlphaMania castmember that is a feathered mask of a person's hair. If you dragged the AlphaMania castmember over the corresponding part of the original image of the person you would be able to use the HSB effect to change the hair color to anything you wanted! This is especially efficient if you are using an 8-bit only AlphaMania castmember.

The Apply drawing method is only really useful in conjunction with one or more effects. Most effects come from Effector Set Xtras available separately from Media Lab, but the Scale functionality built into AlphaMania 2.0 is also implemented as an effect.

### *Apply Drawing Method Considerations*

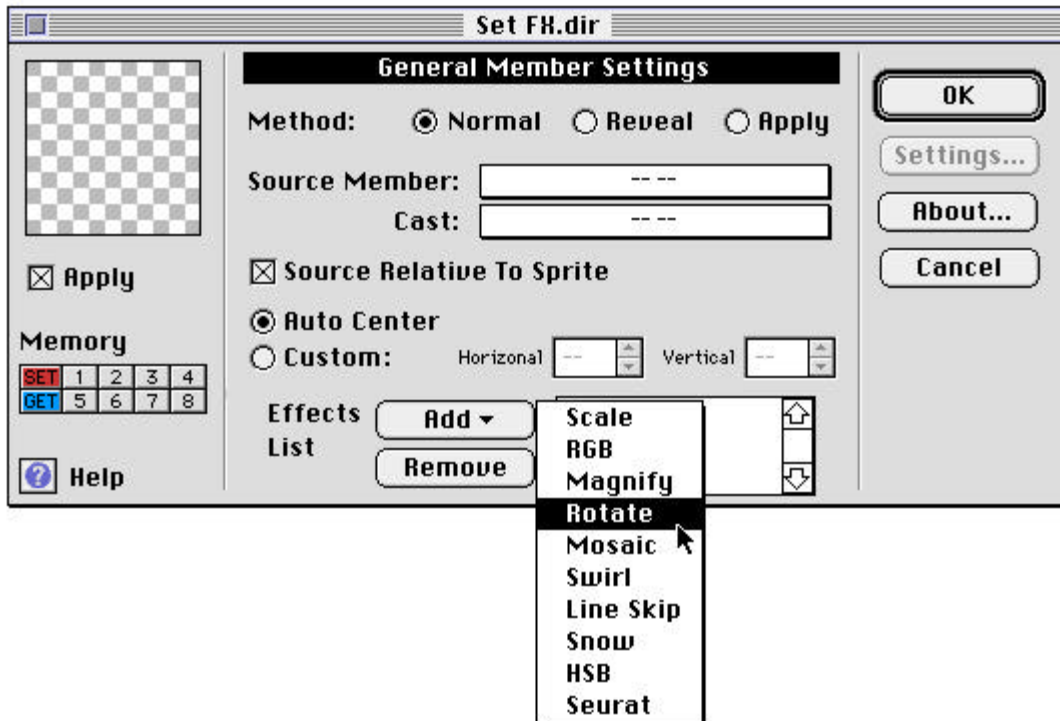
If you have specified the apply drawing method but have not applied an effect to the member or sprite, the sprite will be drawn as a grayscale until you either add an effect to the member or sprite or change the drawing method. It is possible to make the sprite seem to disappear by applying an effect to a background that the effect will not change. For example, if you specify the scale effect using the apply drawing method against a white background the sprite will seem to disappear, because scaling a solid color against itself results in no change to the image.

Because the Apply method affects everything behind it you can use it to treat any area of the stage, no matter what kinds of or how many sprites are stacked there, as if it were an AlphaMania castmember!

### **Using Drawing Methods with the SetFX movie**

#### *Setting the Draw Method*

The SetFX movie can be used to set the Drawing method for a castmember. All sprites created from an AlphaMania castmember use the drawing method specified for that castmember. To manipulate the drawing method, select an AlphaMania castmember in the cast and then select "SetFX" from the Xtras menu. You will be presented with the following dialog (seen on next page):



Using the controls you see here you can select the drawing method to use and you can also set which bitmap castmember is to be revealed when the Reveal method is enabled. Bitmaps you'd like to reveal must be present in the current movie or a currently loaded cast file in order to appear in the list of source bitmaps. You can also set where that bitmap will be displayed, and whether that coordinate is relative to the stage or the sprite. (See "Reveal Method Concepts" above.)

Under Director 5, AlphaMania identifies the source member by cast file number and castmember number. Thus, if you move this member to a different cast position, AlphaMania will be unable to locate it and you will have to specify it again.

Under Director 6 AlphaMania identifies the source member by a global identification number and can find the member even if it has changed positions or cast files. (The member must be located in a cast file that is linked to the current movie.)

Using this window you may also set the sourcePosition and whether that position is relative to the sprite or the stage.

### Using Drawing Methods With Lingo

There are lingo commands to control all aspects of drawing methods.

The drawing method itself is a lingo castmember property called the drawingMethod. The methods are provided as symbols, and so are preceded by '#' signs. To change a member's drawingMethod, simply set this property as you would any other member property. Example:

```
set the drawMethod of member 1 = #normal
```

```
set the drawMethod of member 1 = #apply
set the drawMethod of member 1 = #reveal
set curMethod = #apply
set the drawMethod of member 1 = curMethod
```

The other aspects of drawing methods are also lingo properties. The sourceMember is a castmember property that specifies the bitmap castmember to use with the Reveal drawing method. It can be set at any time, even if the reveal method is not the currently selected drawing method. Its value can be set to any legal reference to a bitmap castmember. Example:

```
set the sourceMember of member 1 = member 5 of castLib "English Graphics cast"
set the sourceMember of member 1 = member "skeleton"
set curSource = member "innards"
set the sourceMember of member 1 = curSource
```

The sourcePosition is also a property which takes a point as its value. If you give it point(65535, 65535) as its value the position will automatically auto-center. Example:

```
set the sourcePosition of member 1 = point(50,50)
set the sourcePosition of member 1 = point(65535, 65535) - - auto-centers
```

You can determine whether the sourcePosition is relative to the sprite or the stage by setting the sourceRelToSprite property of the AlphaMania castmember to true or false. Example:

```
set the sourceRelToSprite of member 1 = true
```

### 8-Bit Only AlphaMania Castmembers

Now that you understand drawing methods, you will see the usefulness of 8-bit only (a.k.a. alpha-only) AlphaMania castmembers. They take up only a quarter of the space on disk and in memory of a standard AlphaMania castmember, yet they are fully as functional using either the Apply or Reveal drawing methods. The only drawback to 8-bit only AlphaMania castmembers is that because they contain no color data they cannot be used with the normal drawing method. Photoshop users can think of them as masks or even adjustment layers.

#### *Importing 8-bit Only Members*

8-bit only AlphaMania castmembers are created by clicking the "8-bit only" checkbox in the AlphaMania import dialog and then choosing a 32-bit file or Photoshop layer. The color data for the file or layer will be ignored and only the alpha channel will be imported.

#### *8-bit Special Considerations*

8-bit only castmembers cannot use the normal drawing method. 8-bit only castmembers are drawn in grayscales in reveal mode when no SourceMember is available, and in apply mode when no effect has been applied. Thumbnails for alpha-only members appear as grayscale masks.

If you would like to determine whether a member is 8-bit or not, use the "alphaOnly" property. It returns true or false. Example

```
if the alphaOnly of member 3 = false then set the drawMethod of member 3 = #normal
```

## Using Effects

AlphaMania 2's most exciting new feature is its ability to use dynamic special effects. Every effort has been made to make the Effector Set effects as easy to use as possible. If you want to jump right in simply select an AlphaMania castmember in your cast and choose the Set FX option from your Xtras menu. This is the quickest way to start using the effects right away. But be sure to come back to this point in the documentation because a firm understanding of the concepts discussed here will make you well armed to really push the effects to their maximum potential.

This section explains the concepts behind effects and how to use them with lingo and with the SetFX movie. The scale effect is the only effect that is included with AlphaMania 2. Other effects are available from Media Lab in the Effector Set Xtras. Many of the examples in this section make use of effects found in the Effector Set Xtras. If you do not own them but would like to explore some of the things they can do, free demos are available from Media Lab's web site.

### **Effect Basics**

At the bottom of every Effector Set effect is an AlphaMania (2.0 or later) castmember or sprite. The Effector Set Xtras in a sense simply endow these castmembers and sprites with new abilities much as the Earth's yellow sun does for Superman. So both an Effector Set and the AlphaMania 2 Xtra must be present to use the effects. (Scale is the only effect included with AlphaMania 2.)

The effects themselves are similar to video effects or animated Photoshop filters with a very important difference: you can control the properties of the effects as the movie plays. You can colorize something, rotate it, etc. all in real time under your control or under the control of the end user of your product.

The simple process of using an effect is as follows:

1. Import an AlphaMania castmember with AlphaMania 2.0 and select it in the cast
2. Add an effect to the castmember using lingo or the provided SetFX movie
3. Adjust the properties of the effect to taste

Steps 2 and 3 can be repeated indefinitely as you add multiple effects with different properties to the same castmember. For example you could make a castmember rotate as it becomes more blue and ripples like water. How? Read on...

### *Common Effect Properties*

While all effects are different from each other, they share many things in common. All effects can animate over time, for example. (See "Animation Modes" below for details.) For the effect to know what to do as time passes you must set starting and ending conditions, well as provide a number of frames for this change to occur across. Many effects can ease-in or ease-out over a number of frames you specify, depending on the animation mode. You can these properties using the SetFX movie in the Xtras menu, or you can set them directly with lingo. The documentation for each

individual effect discusses the unique nature of its own special abilities, but in this documentation you will always see properties with names #startHue, #endHue, #startPercentage, etc.

### *Multiple Effects*

Effects are like stops on an assembly line that exists between your original castmember graphic and how it will look when it is eventually drawn on the stage as a sprite. At the beginning of the assembly line is the source graphic as specified by the base AlphaMania castmember and the drawing method. Each new effect you add to a member or sprite creates a new stop on the assembly line. The graphic travels to each stop, is changed in some way, and then continues to the next stop until all of the effects have done their magic. The graphic is then drawn to the screen.

This means that the effects are cumulative and that changing the order of effects can alter the final result. Each effect manipulates the graphic that resulted from the previous effects. If the first effect makes the graphic more blue, then it is this blue graphic, not the original graphic that is sent to the next effect.

The first effect you add is the first effect applied, and so on. Effects can be removed at any time from any position in the effects order. If you wish to arbitrarily change the order of effects you must remove all of the effects and add them back from scratch in the order you desire.

Note: The more effects you add to a castmember or sprite, the longer your sprite will take to draw. While all of the effects have been highly optimized for speed, too many effects on a large enough castmember can slow even the fastest PowerPC or Pentium to a crawl.

### *Castmember Effects vs. Sprite Effects*

Adding an effect to a castmember using the SetFX movie or lingo will cause that effect to operate on all sprites that are created from that castmember. This is similar to the width of a bitmap castmember. If you change the width of a bitmap member in the paint window, all sprites from this member will change. But if you change the width of a bitmap sprite, only that sprite is affected. When you add an effect to a sprite with lingo, only that sprite is affected.

Similarly, changes made to an AlphaMania castmember's properties with lingo or the SetFX movie are saved with the movie. Changes made to a sprite's properties with lingo are not. (The SetFX movie can't change sprite properties for this same reason.) If you type scale(member 1) into the message window, that information is saved when you save the movie. If you do the same to a sprite while the movie is playing, it will forget as soon as the movie is stopped or the sprite's span ends.

### *Simple vs. Complex Effects*

There is an exception to the rule that effects are applied in the same order they are added. There are two types of effects, *simple* and *complex*. The only difference between them is that complex effects temporarily manipulate the alpha channel of the AlphaMania castmember in the process of applying the effect. This does not change the castmember, but it does mean that these effects must come at the end of the order of effects. AlphaMania manages this limitation for you, but if you notice that you added an effect after another effect and that when you check the effects list the order has reversed, it is because you added a complex effect before a simple effect.

### *Interpolation*

Many effects support interpolation. This is programmer jargon for smoothing. Interpolation slows drawing down a bit, so AlphaMania supports three settings for you to choose from. The first turns interpolation off, which means that an effect will play at its fastest but look a bit worse. The second mode turns interpolation on and makes things look a lot better at the expense of speed. The third mode is a nice between the first two: interpolate when paused. Often when an effect is animating it doesn't need interpolation because it is moving too fast for the eye to detect any blockiness. The third mode turns interpolation off until the effect stops animating. The animation referred to here is effect animation, not score animation. See "Animation Modes" for details.

### *The Sprite Rectangle*

Some effects, namely rotate, can change the size of your sprite rectangle when added. This is done on purpose because the effect will be changing the size of your graphic and is stopping the sprite from cropping. For example with rotate, any non-circular object will take up a varying amount of space as it rotates. When you add rotate to a castmember, it increases the sprite rectangle of sprites made from the castmember so that no matter how you rotate the graphic it will always fit into the new rectangle. (Adding the effect at the sprite level with lingo does not do this and can cause cropping. See the documentation of the Rotate effect for details.)

### *Effects And Behaviors*

Director 6's new drag-and-drop behaviors are incredibly well suited to making complex interactions with effects very easy. We have included a large number of behaviors for you to use and learn from, so be sure to try them out. Behaviors allow complex interactivity with an effect as the movie plays. One of the included examples is a simple game of asteroids with rotating flying meteors and a user-controllable spaceship created entirely from only two behaviors and two bitmaps!

### **Effect Animation Modes**

While all of the effects in Effector Set I are unique there are some common elements to most of them. One of the key ones to understand is the 'animation mode'. All of the effects can be animated in different fashions, and all share some of the four common animation modes: static, range, infinite, and pendulum.

There is a sample movie, 'Methods and Modes' or 'MethMode.dir', that let's user's play with drawing methods and animation modes. Playing with it is a great way to see the different animation modes in action.

**Important:** AlphaMania effect animation modes are independent of score animation. An effect will change over time or not only as you specify using lingo or the SetFX movie. Score animation that makes a sprite move across the stage over a certain number of frames does not affect the animation modes discussed in this section.

### *Static*

In static mode an effect is simply applied at some defined setting and then never changes. There is no animation involved. As an example, think of the rotate effect - in static mode you might specify for it to rotate to 45 degrees. It will immediately redraw angled at 45 degrees and then stay that way.

Static mode is 'low power'. Effects that are in static mode do not eat up any processor time if they aren't moving or having to redraw.

### *Range*

In range mode an effect is applied, changes over time, and then stops. The settings must therefore include a number of frames across which the effect occurs. The effect plays for that number of frames and then becomes static. As an example of this, think again of the rotate effect - in range mode you might specify for it to rotate to 45 degrees across 10 frames. It will then, over the next 10 frames, rotate to an angle of 45 degrees (from whatever angle it started) and then stop.

Once the effect destination is reached, the sprite will stop and behave as if it were in static mode. Most effects can also optionally provide a 'start' point too. So instead of rotate from whatever rotation the sprite starts at to 45 degrees, it could be made to rotate from 120 degrees to 45 degrees. Many effects support additional options like ease in and ease out frames for this mode.

### *Pendulum*

In pendulum mode (sometimes known as Ping-Pong mode) the effect goes out to the specified endpoint and then returns, and then it starts all over again. In pendulum mode the effect runs repeatedly, never stopping. Using rotate as an example, in pendulum mode you might specify 45 degrees and 10 frames. The sprite will then, on the next ten frames, rotate to an angle of 45 degrees (from whatever angle it started), and then across the ten frames after than, swing back to the starting angle, and so on forever.

As with the range animation mode, most effects can optionally provide a start point for this mode, and many effects support relevant option like ease in and ease out frames.

### *Infinite*

In infinite mode the effect goes out to a specified endpoint and then the effect starts over. It does this repeatedly, never stopping. With most effects this will produce a rather jarring effect. Some effects provide a custom implementations of infinite mode. Rotate, for example, when in infinite mode, uses the number of frames is to determine how fast it performs a complete 360 degree rotation. Other effects (like HSB) may also provide custom implementations of this animation mode.

Like pendulum, many effects can optionally provide a start point for this mode, and many effects support relevant option like ease in and ease out frames.

## **Using Effects with the SetFX Movie**

This section discusses the SetFX movie, a Director movie that is included with AlphaMania 2 and should be found in the Xtras menu. If it does not appear there, see "Troubleshooting." This movie is the simplest way to add and modify castmember effects.

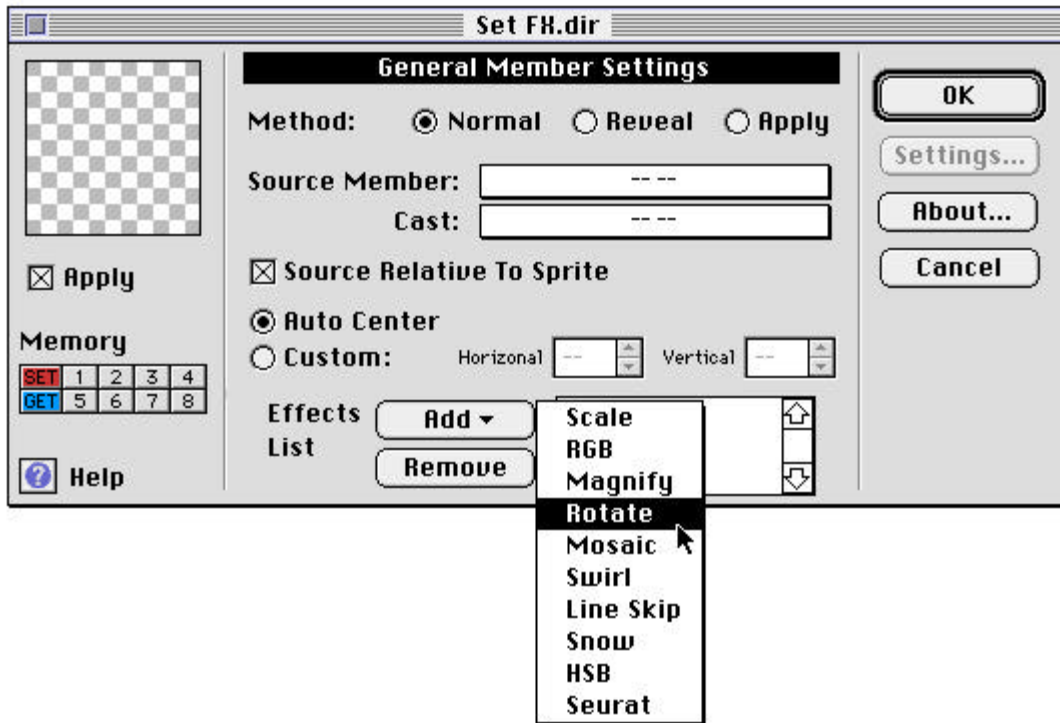


Figure 5. SetFX general settings.

#### *Adding an Effect*

To use the SetFX movie, select an AlphaMania castmember in the cast and select “SetFX” from the Xtras menu. You will be presented with the basic SetFX window that allows you to set and modify the drawing method for the castmember. (See “Drawing Methods”)

Examine the bottom of this window. There are buttons labeled “Add” and “Remove.” You will also see an empty list (assuming you haven’t jumped ahead and added effects already.) Clicking on the Add Button will cause a popup menu to appear which contains all of the effects you currently have installed. (If you only have AlphaMania 2 and no Effector Set Xtras, only Scale will appear in this list.)

When you add an effect from this menu, the name of the effect will appear in this list, and the control panel for this effect will appear in the upper portion of the window. This panel is slightly different for each effect, but all effects share certain attributes.



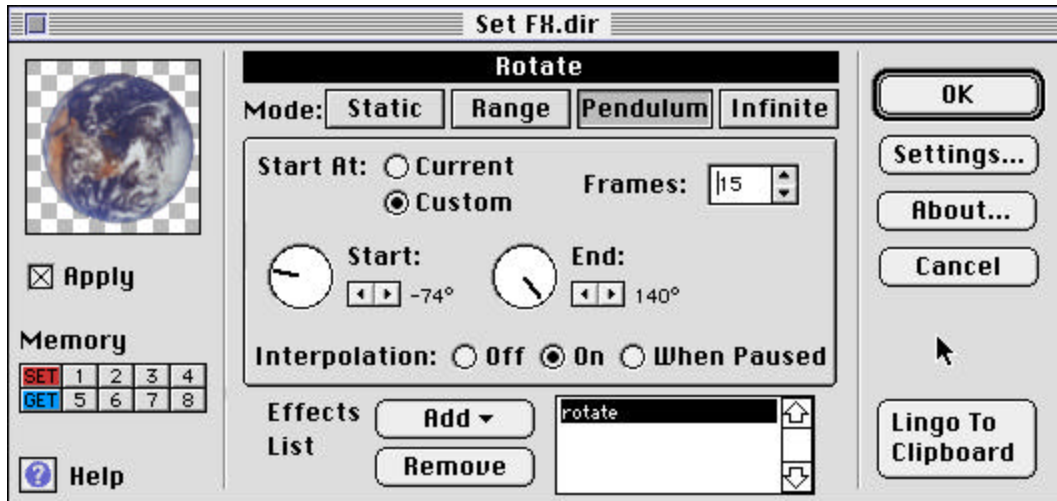


Figure 6. Rotate Settings

### *Editing an Effect*

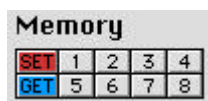
Adding an effect automatically takes you to the appropriate effect editing panel. To edit an effect you have previously added, double-click on the name of the effect in the scrolling list at the bottom of the SetFX window and you will be taken to the appropriate panel.

Once you have reached the panel for controlling the effect you will see buttons at the top of the window that allow you to determine the animation mode for the effect. Clicking on them can cause the control panel to change depending on the animation mode you choose. (See “[Animation Modes](#)”) Within the effect control panel you can set starting and ending properties, number of frames, etc.

### *Effects Preview*

A preview of the settings you choose appears in the preview window at the upper right of the SetFX window. The preview is animated and you can start it over from the beginning by clicking on it.

Underneath the preview is a checkbox labeled “Apply”. This does not mean the apply drawing mode! It causes the current settings to be applied temporarily to any sprites on the stage created from the castmember you have currently selected so that you can preview the effect with your original graphic. The stage preview does not run the animation but can show you the start and end points as you adjust them.



### *The Memory Buttons*

At the left of the window is a number of small buttons labeled “Memory.” These allow you to save and restore effects settings. You can save up to eight effects configurations. To save, click and hold the “Set” button and drag the cursor to the number you would like to assign to the current settings. To restore a configuration, click and hold the “Get” button and drag the cursor to the

number you would like to restore. The configurations that are stored include all effects in the current effects list, not just the effect you are currently editing.

### *Moving Between Effects*

If you have added several effects, to edit any one of them simply double-click on the name of the effect in the effects list at the bottom of the screen. This will take you to the appropriate control panel.

### *Removing Effects*

Find the name of the effect to be removed in the effects list at the bottom of the window and select it. Click the “Remove” button to remove this effect from the effects list.

### *Other Controls*

The “Settings” button takes you back to the panel which allows you to set the drawing method and associated settings for the castmember. The “Lingo to Clipboard” button is available only when editing a specific effect. It copies the lingo command required to create the current effect with the current settings to the clipboard and also puts it to the message window. This can be very helpful for both learning about the lingo used with effects, and as a time saver to actually create your lingo commands.

### *SetFX Special Considerations*

The SetFX movie can only add, move, and manipulate effects at the castmember level. This means that changes you make to a castmember’s effects will be alter all sprites generated from that member. To control effects at the sprite level you can use the behaviors included with the various Effector Set Xtras, or you can write your own lingo.

## **Using Effects with Lingo**

While lingo novices will be able to accomplish quite a bit with simple commands, lingo experts will be able to do almost anything they imagine with the variety and power of the advanced lingo access to effects.

### *Adding Effects*

Basic use of effects with lingo is surprisingly easy. The format for adding and manipulating effects is the same for all effects. Type the name of the effect you would like to create or change, followed in parentheses by the castmember or sprite you wish to modify and a property list containing the information required by the effect. Any arguments that are omitted are given default values or ignored. Most non-numeric arguments are passed in and out as Director symbols, and so are preceded by a “#” sign. Example:

```
scale(sprite 1, [#animMode : #follow] )
- the above causes sprite 1 to scale into the sprites rectangle

rotate(member 4, [ #degrees:45, #interpolation: false ] )
-- the above rotates by 45 degrees all sprites created from member 4
```

The reference manual for the Effector Set Xtra containing the effect you wish to control lists and describes all possible arguments to that effect. It is important to note that, depending on the effect and the mode, certain arguments cannot be omitted. If you issue an effect command to a

castmember or sprite and nothing happens, you probably forgot a required argument. The reference manual entry for an effect will also list what arguments are required for each animation mode, etc. When in doubt, use the SetFX movie “Lingo to Clipboard” command and compare the lingo it produces to your own.

### *Built-In Effect Help (DescribeEffect)*

Within Director you can get a description of the various arguments and animation modes supported by an effect using the describeEffect command. This command must be called with any AlphaMania castmember and the effect you want to know about as arguments. No effect needs to be applied to the castmember you use this command with. The description is returned in the result and may be put to the message window. Example:

```
put describeEffect(member 4, #rotate)
```

### *Checking the Parameters of an Effect (GetEffectArgs)*

Once you have applied an effect, you may wish to check on the arguments you sent to that effect as parameters. The lingo command GetEffectArgs accomplishes this. To use it, pass in an AlphaMania castmember and the name of an effect that has been added to it. Like DescribeEffect it returns its information in the result, which can be put to the message window. Example:

```
rotate(member 1, [#degrees:45, #interpolation: false])
put GetEffectArgs(member 1)
-- [#degrees: 45, #interpolation: 0 ]
```

### *Modifying an Effect After Adding It*

Modifying an effect you have already added to a castmember is simple. AlphaMania automatically keeps track of effects that have been added to a castmember, so when you issue a second command using the same effect, AlphaMania changes the effect you have previously added, rather than creating a new effect. (To create multiple versions of the same effect see the next section.)

Example:

```
rotate ( member 1, [#degrees:45])
put GetEffectArgs(member 1)
-- [#degrees: 45]
rotate [ member 1, [#degrees: 90]]
put GetEffectArgs(member 1)
-- [#degrees: 90 ]
```

### *Managing Multiple Effects*

As the above examples demonstrated, you add an effect to a sprite or member by issuing a command corresponding to the name of the effect and taking the sprite or member as an argument along with optional parameters supplied in a list.

Each time you do this with a different kind of effect, the effect you specify is added to a list of effects that will be performed each time the sprite is drawn.

### Checking the Effects List (GetEffectList)

Once you have added some effects to a sprite or castmember, you can call GetEffectList() on that castmember or sprite to see which effects have been added and in what order. This command returns a lingo list. Example:

```
rgb(member 1, [redShift:50, blueShift:25])
put GetEffectList(member 1)
-- [#rgb : #rgb]
rotate(member 1, [ #animMode : #infinite, #numFrames : 15])
put GetEffectList(member 1)
-- [ #rgb : #rgb , #rotate : #rotate ]
```

### Naming Effects

Notice that the first call returned [ #rgb : #rgb ]. This does not mean that there are two rgb effects applied to this castmember, it means that there is one rgb effect that you have not given a name to, so it took the default name “rgb”. You name an effect by specifying the name when you add the effect. Example:

```
rgb ( member 1, #blueDude, [ blueShift : 50 ] )
put GetEffectList(member 1)
-- [ #blueDude : #rgb ]
```

Now you can refer to the effect by name if you prefer. Example:

```
rgb ( member 1, #colorDude, [ blueShift : 50 ] )
put GetEffectList(member 1)
-- [ #colorDude : #rgb ]
put GetEffectArgs( member 1, #colorDude)
-- [ #blueShift : 50 ]
blueDude(member 1, [ redShift : 20 ] )
put GetEffectArgs( member 1, #colorDude)
-- [ #blueShift : 50, redShift : 20 ]
```

This allows you to add multiple versions of the same effect to a sprite and then refer to them by name.

An important thing to be aware of is that when you add an effect to a sprite that effect is added to the sprite's effect list which already contains any effects which have been added to the castmember. The following example assumes that sprite 1 is created from castmember 1:

```
rgb(member 1, [blueShift:50] )
put GetEffectList(member 1)
-- [#rgb : #rgb]
rotate( sprite 1, [#animMode : #infinite, #numFrames : 15] )
put GetEffectList(sprite 1)
-- [#rgb : #rgb, #rotate : #rotate]
-- the sprite now has two effects in its list
put GetEffectList(member 1)
-- [#rgb : #rgb]
-- the member still has just one effect
```

Even though the `GetEffectList` command returns a lingo list, modify this list directly with the standard lingo list commands does not affect the effects themselves. It is simply for your information.

### *Removing Effects (RemoveEffect, RemoveAllEffects)*

If you need to remove an effect from the effect list for a sprite or castmember, use the `RemoveEffect` command. It takes the castmember or sprite as the first argument and the name of the effect as the second argument. Example:

```
rgb(member 1,[blueShift:50])
put GetEffectList(member 1)
-- [ #rgb ]
rotate(member 1, [#animMode :#infinite, #numFrames:15])
put GetEffectList(member 1)
-- [ #rgb : #rgb, #rotate : #rotate ]
removeEffect( member 1, #rotate )
put GetEffectList(member 1)
-- [ #rgb : #rgb ]
```

If you want to remove all of the effects from a sprite or castmember, use the `RemoveAllEffects` command. Example:

```
removeAllEffects( member 1 )
```

### *Custom Effect Functions*

Many effects provide custom functions to aid lingo programmers in controlling the effects' unique special capabilities. Any effect that interpolates will support two custom functions, `InterpolateNow` and `SetInterpolation`, which allow you a more convenient and precise level of control over interpolation. The rotate effect in particular supports a wealth of custom functions which make common rotation programming tasks much easier. The documentation for each effect describes the custom functions unique to that effect.

## Part 7: The Scale Effect

### *Scale Basics*

Effect Type: Complex

The scale effect brings to AlphaMania what traditional Director bitmap sprites have had for a long time: the ability to stretch and change size. But unlike Director bitmaps, AlphaMania castmembers look GREAT when they scale, and have many more controls over their scaling.

### **Custom Animation Modes**

#### *Follow*

Scale, along with the traditional animation modes (static, range, pendulum, and infinite) also introduces the follow mode. In the follow mode, scale simply follows the sprite rectangle and will stretch to fit the rectangle size. Optionally, the percentage parameter can be used to make the sprite scale to a percentage of the rectangle. This is useful when combining the scale effect with other effects that manipulate the sprite rectangle (like the rotate effect in Effector Set I).

The follow mode is the default mode for the scale effect. So simply calling `scale(sprite x)` will put the sprite into follow mode with interpolation on and is exactly like calling `scale(sprite x, [ #animMode : #follow ] )`

### **Custom Functions**

#### *Follow*

```
Follow( sprite x, #scale)
Follow( sprite x, #scale, interpolation)
Follow( sprite x, #scale, interpolation, percentage)
```

The Follow function will put the scale effect into follow mode immediately.

#### *InterpolateNow*

```
InterpolateNow( sprite x, < position or #effectSymbol or #customName >)
```

This function causes the sprite to immediately redraw interpolated, but does not change the flag for the sprites overall interpolation setting.

#### *SetInterpolation*

```
SetInterpolation(sprite x, < position or #effectSymbol or #customName >, interpolation)
```

## Scale Reference

Effect Symbol: #scale

Properties	Type	Value	Default	Exclusive
#animMode	symbol	#static, #range, #pendulum, #infinite, #follow	#follow	
#numFrames	integer	1+	-	required in infinite, range, and pendulum
#interpolation	integer	0, 1, 2	1	-
#percentage	integer	1+, 100 = no scale	100	
#startPercentage	integer	1+, 100 = no scale	current	invalid in follow and static
#endPercentage	integer	1+, 100 = no scale	#percentage	
#width	integer	any	-	invalid in follow mode
#height	integer	any	-	invalid in follow mode
#startWidth	integer	any	current	invalid in follow and static
#startHeight	integer	any	current	invalid in follow and static
#endWidth	integer	any	#width	
#endHeight	integer	any	#height	

### Custom Modes

#### *#Follow*

The #follow mode is a custom mode for the scale effect only. In this mode, the AlphaMania sprite will scale to the current sprite rectangle. This mode replicates the standard Director sprite scaling. Optionally, #follow mode can be set up to scale based on a percentage of the sprite rectangle by using the #percentage parameter.

### Custom Functions

#### *Follow*

```
Follow( sprite x, < position or #effectSymbol or #customName > )
Follow( sprite x, < position or #effectSymbol or #customName >, interpolation)
Follow( sprite x, < position or #effectSymbol or #customName >, interpolation,
percentage)
```

This is the mode accessor function for accessing the 'follow' mode.

#### *InterpolateNow*

```
InterpolateNow( sprite x, < position or #effectSymbol or #customName > )
```

This function causes the sprite to immediately redraw interpolated, but does not change the flag for the sprites overall interpolation setting.

### *SetInterpolation*

SetInterpolation(sprite x, < position or #effectSymbol or #customName >, interpolation)

### **Arguments**

#### *interpolation*

This argument sets the interpolation setting for this effect. With interpolation on the scaled sprite will be smoother and nicer looking, but draw slower. With interpolation off, the sprite will draw faster, but be rougher. There is a get-best-of-both-worlds interpolation setting that will make the sprite only interpolate when paused. A pause state occurs whenever the sprite is in #static or #follow modes, when in #range mode after it has finished the animation, or when the effect has been paused with a call to PauseEffect or PauseAllEffects. Note, the normal lingo command 'pause' has no effect on the pause state of this effect.

---

#### Settings:

Interpolation Off:	0
Interpolation On:	1
Interpolate When Paused	2

---



*percentage*

In all the standard animation modes this argument sets a destination size based on a percentage relative to the default cast rectangle. In #follow mode this argument sets a size that will be continually calculated as a percentage of the current sprite rectangle. Valid values are from 1 up, 100 being the 'normal'. The percentage argument cannot be used at the same time as the width and height arguments.

*width, height*

The width and height are measured in pixels, and the effect will scale to that dimension. These parameters cannot be used at the same time as the percentage parameter, nor will they work with the follow animation mode. If only one is used, the other will default to the current height or width.

*startWidth, startHeight, startPercentage*

These parameters set the starting point for any of the animating modes. If not used, the animations will begin at their current settings, whatever that may be. Again, the percentage arguments cannot be used at the same time as the width and height arguments.

*endWidth, endHeight, endPercentage*

These parameters set the ending point for any of the animating modes. Technically, they are redundant versions of the width, height, and percentage parameters, and aren't needed. They are provided to ease the reading of lingo code.

## Part 8: Register, Purchase, Distribute, Contact Info

### How to Register AlphaMania

AlphaMania is distributed as an unlockable demo, which can be unlocked with a code number which you receive at time of purchase. The demo version of AlphaMania defaces the imported graphics with red lines. See Where To Find AlphaMania below for common locations to find up to date versions. Download the demo to try it out and then call in to order.

To purchase the produce and deactivate the defacing function, you will first need to determine your product ID. To do this:

1. Install AlphaMania into Director 5's Xtras folder and launch Director 5.
2. Click the "About" box.

In the About box for AlphaMania you will find the 7-digit AlphaMania product ID. With that number handy, simply phone us (***1-800-282-5361***) to place your order for AlphaMania and we'll give you your product unlocking code. Or you can order via fax or email by filling out the order form that comes with the AlphaMania downloadable version, being sure to include the Product ID number. Enter that unlocking code into the "unlock" field in the About box and you're done! Keep that code safe somewhere in case you have to re-install AlphaMania or Director for any reason.

**IMPORTANT:** AlphaMania locks itself to the copy of Director in which it is installed. Moving it to a different copy of Director with a different serial number will cause it to revert to "demo" mode, reactivating the defacing function. So make sure you are getting your product ID from the copy of Director you expect to be using AlphaMania with.

### Purchasing

The easiest way to purchase the on-line version of AlphaMania is with a credit card via the automated sales system on our Web page. ([www.medialab.com](http://www.medialab.com)) Your card will be processed immediately, and you will be issued an unlock code.

If you prefer to talk to a real person, or have any problem ordering via the Web site, call ***1-800-282-5361*** to purchase AlphaMania using your Visa or MasterCard, at which time you will be given an unlock code. Alternatively, you may order by email or fax and receive the code number via electronic mail. If you are ordering AlphaMania without having already downloaded and installed a demo version, we encourage you to have the last 10 digits of your Director 5 serial number ready when you call. This is not required, but will mean you won't have to call back for an unlock code once you receive the product.

**Ordering by Phone**

In the U.S: 800.282.5361  
International: +1.303.499.5411

Orders can be taken Monday through Friday, 9 AM to 5 PM Mountain Standard Time.

**Ordering by E-Mail**

*xtra-sales@medialab.com*

There is an order form (ORDER.TXT) provided with the electronic download. Fill in all the information it requests, including your Product ID (see above), and e-mail back to *xtra-sales@medialab.com*

**Ordering by Fax**

+1.303.665.0827

There is a order form (ORDER.TXT) provided with the electronic download. Fill in all the information, including your Product ID (see above), and fax it back to us.

**Distributing AlphaMania**

AlphaMania must be distributed with your projectors that contain AlphaMania cast members. There is no license fee for distributing AlphaMania with your products. The AlphaMania Xtra must reside in a folder named "Xtras" that appears in the same folder as your projector.

To use AlphaMania in your ShockWave movies, the user must have ShockWave for Director 5 and must have AlphaMania in the Xtras folder that appears in the same folder that their ShockWave plug-in does.

**Contacting Media Lab, Inc.**

Media Lab, Inc.

400 S. McCaslin Blvd.

Louisville, CO 80027 USA

Phone: 800.282.5361 to purchase PhotoCaster  
+1.303.499.5411 all other calls

Fax +1.303.665.0827

Website: <http://www.medialab.com/software/>

Distribution Inquiries: [sbruce@medialab.com](mailto:sbruce@medialab.com)

Press and Advertising: [sbruce@medialab.com](mailto:sbruce@medialab.com)

email: [xtra-help@medialab.com](mailto:xtra-help@medialab.com)

[xtra-sales@medialab.com](mailto:xtra-sales@medialab.com)

[xtra-bugs@medialab.com](mailto:xtra-bugs@medialab.com)

Technical support.

Place orders for PhotoCaster

Report any bugs or request new features.

## Part 9: Legal Stuff

### Licensing

When you purchase AlphaMania, you are purchasing the license to use AlphaMania for authoring with a single copy of Director on a single machine. This fee also covers distribution of the Xtra with your Director and ShockWave projects. There is no fee associated with distributing AlphaMania for playback only.

### Copyright Information

AlphaMania ©1996 As Is Software, Inc. AlphaMania is distributed by Media Lab, Inc.. Any distribution related inquiries should be emailed to sbruce@medialab.com.

### Trademarks

AlphaMania and PhotoCaster are trademarks of Media, Lab, Inc. Adobe Photoshop is a trademark of Adobe Systems, Inc. Director and xRes are registered trademarks of Macromedia, Inc. Macintosh and MacOS are registered trademarks of Apple Computer, Inc., and Windows 95 and Windows NT are trademarks of Microsoft Corporation. Electric Image is a trademark of Electric Image Inc.

### License Agreement

You should carefully read the following terms and conditions before proceeding. By choosing the "Agree" button below you are agreeing and indicating your acceptance of these terms and conditions. If you do not agree with them you should choose the "Disagree" button below and the Program will not proceed .

Media Lab, Inc. (hereinafter referred to as Company), a Colorado based Corporation, provides the AlphaMania software (hereinafter referred to as Program) contained on the medium in this package and licenses its use. You assume full responsibility for the selection of the Program to achieve your intended results and for the installation, use and results obtained from the Program.

#### *License*

This license agreement applies to the unlocked AlphaMania software only. You are allowed to distribute the demo AlphaMania software freely. You are prohibited, however, from giving your unlock code to any other user or from using the unlocked Program on more than one computer simultaneously. The unlocked Program is licensed to be used with a single copy of Macromedia Director.

A. In consideration of the payment of a license fee, you are granted a personal, nontransferable and nonexclusive license to use the Program under the terms stated in the Agreement. You own the diskette or other physical media on which the Program is provided under the terms of this Agreement, but all title and ownership of the Program and enclosed related documentation (hereinafter referred to as Documentation), and all other rights not expressly granted to you under this Agreement, remain with the Company.

B. The Program may be used by you only on a single computer. Therefore, you must treat this software just like a book. With the exception of Item D below, just like a book means that this software may be used by any number of people and may be freely moved from one computer location to another, so long as there is no possibility of it being used at one location while the same copy is at the same time being used at another location. Just as the same copy of a book cannot be read by two different people in two different places at the same time, neither can the same copy of software be used by two different people in two different places at the same time. Contact Media Lab, Inc. for details.

C. You and your employees and agents are required to protect the confidentiality of the Program. You may not distribute or

otherwise make the Program or Documentation available to any third party.

D. You may not copy or reproduce the Program or Documentation for any purpose except to make one (1 ) archival copy of the Program, in machine readable or printed form for back up purposes only in support of your use of the Program on a single computer. You must reproduce and include the Company copyright notice on the backup copy of the Program.

E. Any portion of the Program merged into or used in conjunction with another program will continue to be the property of the Company and subject to the terms and conditions of this Agreement. You must reproduce and include Company's copyright notice on any portion merged in or used in conjunction with another program.

F. You may not sublease, assign or otherwise transfer the Program or this license to any other person without the prior written consent of Company.

G. You acknowledge that you are receiving on a LIMITED LICENSE TO USE the Program and Documentation and that the Company retains title to the Program and Documentation. You acknowledge that Company has a valuable proprietary interest in the Program and Documentation. You may not use, copy, modify or transfer the Program or Documentation or any copy, modification or merged portion in whole or in part except as expressly provided for in this Agreement. If you transfer possession of any copy modification or merged portion of the Program or Documentation to another party, your license is automatically terminated.

### *Term*

This license granted to you is effective until terminated. You may terminate it at any time by returning the Program and Documentation to Company together with all copies, modifications and merged portions in any form. The license will also terminate upon conditions set forth elsewhere in the Agreement, or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to return the Program and Documentation to Company together with all copies, modifications and merged portion in any form. Upon termination, Company can also enforce any rights provided by law.

The provision of this Agreement which protects the proprietary rights of Company will continue in force after termination. Termination of this license either voluntary or involuntary does not entitle you to a refund of your purchase cost except as provided elsewhere in this License Agreement.

### *Limited Warranty*

Company warrants, as the sole warranty, that the medium on which the Program Is furnished will be free from defects in materials and workmanship under normal use and conditions for a period of ninety (90) days from the date of delivery to you as evidenced by a copy of your receipt. No distributor, dealer or any other entity or person is authorized to expand or alter either this warranty or this Agreement. Any such representations will not bind the Company. Company does not warrant that the functions contained in the Program will meet your requirements or that the operation of the Program will be uninterrupted or error-free.

Except as stated above in this section, the Program and Documentation are provided as is without warranty of any kind either expressed or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. You assume entire risk as it applies to the quality and performance of the Program and Documentation. Should the Program prove defective you and not Company, authorized Company Distributor or dealer, assume the entire cost of all necessary servicing repair or correction. This warranty gives you specific legal rights and you may also have other rights which vary from state to state. Some states do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

### *Limitation of Remedies*

Company's entire liability and remedy will be:

A. The replacement of any medium not meeting Company's Limited Warranty explained above and which is returned to Company or an authorized Company distributor or dealer with a copy or your receipt; or

B. If Company is unable to deliver a replacement medium which conforms to the warranty provided under this Agreement, you may terminate this Agreement by returning the Program and Documentation to Company, authorized Company Distributor, or dealer from whom you obtained the Program and your license fee will be refunded.

### *Product Returns*

If you must ship the Program and Documentation to an authorized Company Distributor, dealer or to Company, you must

prepay shipping and either insure the Program and Documentation or assume all risk of loss or damage in transit. To replace a defective medium during the ninety (90) day warranty period, if you are returning the medium to Company, please send us your name and address, the defective medium and a copy of your receipt at the address provided below. In no event will Company be liable to you for any damages direct, indirect, incidental or consequential, including damages for any lost profits, lost savings or other incidental or consequential damages arising out of the use or inability to use such Program and Documentation, even if Company has been advised of the possibility of such damages or for any claim by any other party. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages so the above limitation or exclusion may not apply to you. In no event will Company liability for damages to you or any other person ever exceed the amount of the license fee paid by you to use the Program regardless of the form of the claim.

#### *US Government Restricted Rights*

The Program and Documentation are provided with restricted rights. Use, duplication or disclosure by the US Government is subject to restrictions as set forth in subdivision (b) (3) (ii) of the Rights in Technical Data.

#### *General*

This Agreement is governed by the laws of the state of Colorado (except federal law governs copyrights and register trademark(s)). If any provision of this Agreement is deemed invalid by any court having jurisdiction, that particular provision will be deemed deleted and will not affect the validity of any other provision of this Agreement. Should you have any questions concerning this Agreement, you may contact Media Lab, Inc. at the address below.

Media Lab, Inc., 400 S. McCaslin Blvd., #211, Louisville, CO 80027, Phone: (303) 499-5411, Fax: (303) 665-0827  
<http://www.medialab.com>