

# The Graphics Supermarket for the Apple IIGS



# The Graphics Supermarket for the Apple IIGS

Designed and written by Rodger Smith

Program Art by Joe Hebling

Manual by Rodger Smith, Mary Carol Smith, Connie Holvey



P.O. BOX 2440  
EUGENE, OR 97402

## **System Requirements**

Apple IIGS with 512K of ram (768K or more is best)

Color Monitor (Analog RGB is best)

One or more disk drives

Mouse

Product Parts Checklist: 3.5" disk (optional source-code disk), Manual

Published by  
Abracadata  
P.O. Box 2440  
Eugene, OR 97402  
503/342-3030

## **Copyright**

Copyright (c) 1988 Abracadata, Ltd. All rights reserved. No part of this book or the program contained in this package may be reproduced or transmitted in any form or by any means, electronic, or mechanical, including photocopying recording, or by any information storage and retrieval system, without explicit permission in writing from the copyright owner and publisher.

Abracadata and Graphics Supermarket are trademarks of Abracadata, Ltd. Apple IIGS is a trademark of Apple Computer, Inc.

# Table of Contents

<b>I. Introduction</b> .....	7
A. System Requirements .....	7
1. Minimum System .....	7
2. 512K Limitations .....	7
B. Getting Started .....	7
C. About This Manual and Graphics Supermarket .....	8
1. Screen Display And Terminology .....	8
2. Dialog Boxes .....	11
D. How To Try Out Graphics Supermarket .....	12
1. How To Draw .....	13
2. How To Use Color .....	13
3. How To Load From Disk .....	15
4. How To Save To Disk .....	16
5. How To Change Image Sizes .....	16
6. How To Use The Clipboard .....	16
7. How To Change Modes .....	17
8. How To Print Cards .....	17
<b>II. Clipboard</b> .....	19
A. Hold .....	19
B. Display - Edit .....	20
C. Paste .....	20
1. Pasting a Clipboard Image .....	20
2. Other Uses for Paste .....	21
D. Release .....	22
E. Release All .....	22
F. Animation Test .....	23
<b>III. Status</b> .....	20
A. Byte Width and Byte Height .....	24
B. Pixel Width and Pixel Height .....	24
C. Image Byte Size .....	24
D. Clipboard Byte Size .....	24
E. Total Ram Size .....	25
F. Ram Free .....	25
G. Ram Used .....	25
H. Largest Block Free .....	25
I. Managing Memory .....	25
1. Memory Fragmentation .....	26
2. Memory Restrictions .....	26
3. Clipboard Memory Management .....	26
4. If You Run Out of Memory .....	27
<b>IV. Graphic Modes</b> .....	28

<b>V. Drawing Tools</b> .....	<b>29</b>
A. Edit Menu .....	29
1. Undo .....	29
2. Erase Edit Area .....	29
3. Erase Display Area .....	30
4. Erase Screen .....	30
5. About Image Sizes .....	30
6. Drag Frame To Size .....	30
7. Input Frame Size .....	31
8. Full Screen Size .....	31
9. Move Frame .....	32
10. Hide Frame - Show Frame .....	32
11. Horizontal Flip .....	32
12. Vertical Flip .....	32
13. Scroll Edit Area .....	33
14. Scroll Parameters .....	33
15. Hide Icons - Show Icons .....	34
B. Penseize Menu .....	34
C. Draw Menu .....	34
1. Magnify .....	34
2. Erase .....	36
3. Sketch .....	36
4. Spray .....	36
5. Fill .....	37
6. Line .....	37
7. Rectangles / Ovals / Arcs .....	37
<b>VI. Color Tools</b> .....	<b>39</b>
A. Color Palette .....	39
1. The Palette .....	39
2. The 320 Mode Color Palette .....	39
3. The 640 Mode Color Palette .....	40
B. Color Icon Tools .....	41
1. Draw Color .....	41
2. Background Color .....	41
3. Red, Green, and Blue icon tools (RGB) .....	42
4. Swapping and Copying Colors .....	44
5. Blending Colors .....	48
<b>VII. Disk Loads And Saves</b> .....	<b>49</b>
A. Disk Saves .....	49
1. Save Dialog .....	49
2. Picture Saves .....	50
3. Clipboard Saves .....	50
B. Disk Loads .....	51
1. Load Dialog .....	51
2. Picture Loads .....	52
3. Clipboard Loads .....	52
C. Deleting Files .....	53

<b>VIII. Printing Images</b> .....	<b>54</b>
A. About Printing .....	54
1. Menu Items For Printing .....	54
2. Memory Requirements For Printing .....	54
3. Control Panel .....	55
B. Choose Printer .....	55
C. Page Setup .....	56
1. Paper .....	57
2. Vertical Sizing .....	58
3. Printer Effects .....	58
4. Orientation .....	58
5. Printer Effects .....	59
6. Reduce or Enlarge .....	59
D. Print .....	59
1. Quality .....	60
2. Page Range .....	60
3. Copies .....	60
4. Paper Feed .....	61
5. Color .....	61
6. Pages .....	62
7. Copies .....	62
8. Paper Source .....	62
9. To Start Printing .....	62
<b>IX. Programming Images</b> .....	<b>63</b>
A. Generating Source Image Files .....	63
1. How To Save Source Code .....	63
B. Generating Binary Image Files .....	65
C. File Formats .....	66
1. Source Code File Format .....	66
2. Binary File Format .....	69
3. Programming Examples .....	72
<b>X. The POKE That Should Be In The IIGS Control Panel</b> .....	<b>87</b>
<b>XI. Source Code Disk Description</b> .....	<b>89</b>
<b>Index</b> .....	<b>91</b>

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

# I. Introduction

## A. System Requirements

### 1. Minimum System

Apple IIGS with 512K of ram (768K or more is best)  
Color Monitor (Analog RGB is best)  
One or more disk drives  
Mouse

### 2. 512K Limitations

If your system has only 512K of memory, you can still use *Graphics Supermarket* very effectively. However, you need to manage memory more carefully.

You can still print images (in color with an Imagewriter II and color ribbon), but you don't have as many printing options as you do with more memory. The *Printing Images* section describes the standard options available with 768K (or more) of memory, and indicates which ones aren't available with a 512K system.

You will still find the Clipboard very useful with 512K, however, you can't hold an image larger than about half the screen to the Clipboard. With a 512K system, the largest free block of memory is about 16K. A full screen (and the data saved with it) would require slightly more than 32000 (32K) bytes of contiguous memory. You have more than 16K of free memory in a 512K system, and you can still hold up to 15 smaller images to the Clipboard. But the maximum size of any one image you can hold is determined by the largest block of contiguous free memory.

Be sure to read the *Status* section to see how to keep tabs on available memory, and how to manage it.

All the other *Graphics Supermarket* features are available on 512K systems.

## B. Getting Started

First, make a backup copy of your original *Graphics Supermarket* disk. Use the "FASTCOPY" or other copy program on your GS system disk. Follow the on-screen instructions. Then put your original disk in a safe place and use the backup for everyday use. If you damage the backup, you can make another one from the original.

You also need to create a Prodos data disk to store images you wish to save. Format a blank disk, or use a previously formatted disk that has plenty of room for images. Be sure to remove the original *Graphics Supermarket* disk from all drives during any formatting process or you may permanently damage it. Use your GS system disk for formatting also. If you need help, refer to your *Apple IIGS System Disk User's Guide*.



Remember, *Graphics Supermarket* requires a minimum of 512K to load (more is better). If you are using a ram disk, leave 768K of ram free to get the best performance. Consult Appendix A of your *Apple IIGS Owner's Guide* and Chapter 2 of the *Apple IIGS memory Expansion Card Owner's Guide* to see how to access the Control Panel and set the ram disk size. (If you change the ram disk settings, you need to turn the computer off and re-boot before the changes take effect.) If you have not previously set up a ram disk, this doesn't apply to you. All of your ram is free and available for use (the *Printing Images* section of this manual shows how to set the Control Panel for printing images).

To get started, insert your copy of the *Graphics Supermarket* into your disk drive, turn your computer off, wait 15-30 seconds, and turn it back on. *Graphics Supermarket* loads and starts automatically (after a little wait while the operating system, the program, the tools, and the printer drivers load). After they load, you see a help message. Read the message, then use the mouse to move the arrow onto [Ok]; press the mouse button.

## C. About This Manual and Graphics Supermarket

Even if you have used drawing programs before, *Graphics Supermarket* has unique features related to image sizes, the Clipboard, and the formats you can save images in (such as source code). Read the manual completely to get the most out of these and other features. The *Graphics Supermarket* has a lot of flexibility and power, and you'll be able to use it when you become completely familiar with it.

### 1. Screen Display And Terminology

#### a. Mouse And Cursor

*Graphics Supermarket* lets you use the mouse to make selections and to draw.

When you move the mouse, an arrow or small dot also moves on the screen. We refer to the arrow or dot as a *cursor*, or *pointer*, in this manual.

The size and shape of the cursor can change. When it's not an arrow, it will most often be the *Edit Cursor* used for drawing. The Edit Cursor looks like a small dot, or at larger sizes, a small square. The Edit Cursor indicates where a draw operation will take place on the screen.

When you move the cursor to the drawing area on the screen, it changes from the arrow to the Edit Cursor. When you move to where you might make a menu bar or icon bar tool selection, it changes back to an arrow.

#### b. Click And Drag

To *click* means to quickly press and release the mouse button. A *double-click* is two quick presses of the button.

To *drag*, press and hold down the mouse button and move the mouse. Release the mouse button to stop *dragging*. You may occasionally need to drag when making a selection or using some of the drawing tools (like drawing lines).

#### **c. Menu Bar**

The Menu Bar is a horizontal list of menu titles that always appears at the top of the screen. The titles include FILE, EDIT, PENSIZE, DRAW, MODE, COLOR, and STATUS. In the 320 X 200 graphic mode (the start up mode), the menu titles are shortened to 3 letters so they will fit on the screen. For example, FILE is shortened to FIL.

To see a menu, use the mouse and move the cursor to a title, then press and hold down the mouse button; you'll see a list of menu choices below the title. This manual refers to menu choices as *menu items*, *items*, or *pull-downs*.

To select a menu item, keep the mouse button pressed and move the mouse, or the arrow, downward. When the item you want is highlighted (white letters on black), release the mouse button. The menu item will flicker a few times indicating it was selected.

#### **d. Icon Bar**

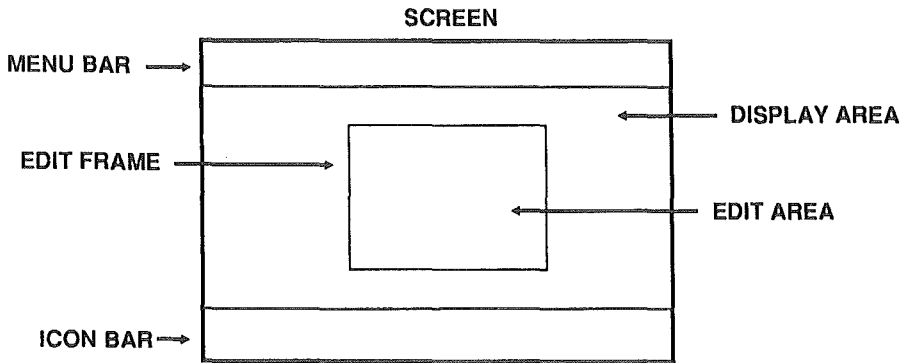
The *Icon Bar* is the display across the bottom of the screen. The shapes and symbols are called *icons*. They represent drawing and color tools which you can use to create images. See the *Drawing Tools* and *Color Tools* sections for complete descriptions.

Many of the options on the Icon Bar are the same as in the pull-downs. Use the Icon Bar to select tools faster, easier, and for some special tools.

#### **e. Edit Area**

The *Edit Area* is a specific area on the screen where you draw or edit images. You'll recognize it as "the area inside the frame". You can always change the size of the Edit Area. See a brief demonstration of how to change the size below in the *Introduction* section, and a complete description is in the *Drawing Tools* section.

The important thing to remember is that you can only draw inside the Edit Area. Read on for more related information.



#### f. Edit Frame

The Edit Area (described above) is enclosed by an *Edit Frame*. You only see the Edit Frame if the Edit Area is small enough for the frame to fit on the screen. When you start up *Graphics Supermarket*, the Edit Area is set to full screen size, so the Edit Frame is not visible. Later in this section, smaller image sizes are discussed.

Since you can change, or specify image sizes by re-sizing the Edit Frame, you can create a series of images exactly the same size, which is especially useful if you are a programmer drawing clip art, or for creating images in programs.

#### g. Display Area

The *Display Area* is the area outside the Edit Frame (or Edit Area). At start up, no display area exists since the Edit Area is the entire screen.

#### h. Image Size

The *Image Size* is the same size as the Edit Area [and you already know, if you read e and f (above) that this area is surrounded by a frame; unless it's a full-screen image, in which case you cannot see the frame]. You can change the Image Size by dragging the Edit Frame to a new size, or by entering a new size (see *Drawing Tools*). Use the **STATUS** pull-down to check the current Edit Area or image size. These measurements refer to the area inside the edit frame, not to the frame itself.

#### i. Draw Color

This term indicates the currently selected or active drawing color. The large square at the right end of the color choices (or color palette) shows the currently selected *Draw Color*.

#### j. Background Color

*Background Color* refers to the color of the screen when it is completely erased. All erasing is done with the Background Color. The Background Color is normally the

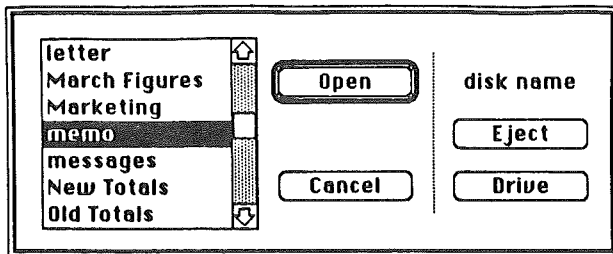
color you draw images on top of. See the *Color Tools* section on how to change the background color.

### k. Pixels

A *pixel* is the smallest screen dot possible in a given graphic mode, or put another way, the smallest dot that you can draw is 1 pixel. The 320 X 200 super high-resolution graphic mode refers to a screen that's 320 pixels wide and 200 pixels high. 640 X 200 means you have 640 pixels across and 200 pixels up and down.

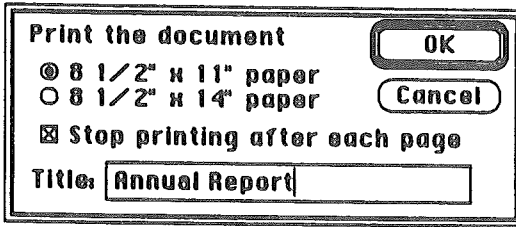
## 2. Dialog Boxes

A *dialog box* is one of the ways you communicate with the *Graphics Supermarket* program. It lets you make selections and provides a way for you to tell *Graphics Supermarket* how you would like something done. Most of the time you use your mouse to make dialog box selections, but occasionally you may need to type some information. Sometimes you select names in a list -- for example, a filename for a disk load. To select a name, move the cursor over it and click. The currently selected name is highlighted.



### a. Dialog scroll bars

The scroll bar is a vertical bar along the side of a list of file names (or other data) inside a dialog box. If there isn't room to show all the names at the same time, you can either use arrows to scroll the list and see the other names, or you can move the scroll bar (or box), located between the arrows. Simply click on the arrows to scroll, or hold down the mouse button for continuous scrolling. To use the scroll bar, move the cursor to it, hold down the mouse button, and drag the bar upward or downward. The size of the bar gives you a clue as to how long your list is. A short bar means a long list of names or data is not visible. A long bar that allows you only a little movement means there are only one or two files not visible. If the scroll bar doesn't move, it means you're looking at all the available files.



### b. dialog options

Many dialogs have options you can turn on or off, that is, enable or disable. For example, the print dialog has a color option. Options may have round buttons or square boxes beside them. A filled button or a box with an [X] inside it means the option is selected, or enabled. To select an option, click on the round button or box beside it. Some options work like a switch; click them a second time to disable them.

### c. dialog text edit fields

Some dialog boxes require you to type information, such as a name or number. These usually have one or more *edit field* rectangles where your typing appears. You can edit the text you have typed by using delete. If there are several edit fields (rectangles) in a single dialog, press [tab] to move from one to another.

### d. dialog buttons

You may be familiar with dialog [Ok] and [Cancel] buttons. Often, there are other kinds of buttons as well. The name inside a button indicates what action will be taken when you select it.

A double border enclosing a button indicates it's the default button. The default button is selected if you press [Return]. To select a button, move the cursor over it and click.

## D. How To Try Out Graphics Supermarket

This section introduces some of the features you can use to draw, to print holiday cards that are included on disk, or load and save images to disk for permanent storage, or to the Clipboard for temporary storage. It's best to read and do the tryouts in sequence.

The examples and directions are brief and introductory only. Detailed descriptions are given in other sections of the manual.

First boot up your copy of *Graphics Supermarket*. When you get to the main screen, your edit area is the entire screen.

# 1. How To Draw

You can select the drawing tools described in this section from the Icon Bar, or from the **PENSIZE** and **DRAW** menus. See the *Drawing Tools* section for complete descriptions.

## a. Sketch

Click on the pointed end of the pencil icon (top row, third icon from the left). Move to the Edit Area (drawing area). To *sketch*, press and hold down the mouse button while moving the mouse. Release the button to stop.

## b. Lines

Select the slanted line icon (top row, sixth icon from the left). Move to the line's start location. Press and hold down the mouse button. Move the mouse to drag the other end of the *line* to the desired location and release the button to leave the line there (draw it).

## c. Pensize

To change the width of a line, go to the **PENSIZE** pull-down. Press and hold down the mouse to see the menu items. A check mark beside an item shows the currently selected size. Move the mouse downward and highlight the largest size. Release the mouse button to select it. Try drawing a line with the new *pensize*; experiment with other sizes too.

You can also click a pensize icon on the Icon Bar to change the pensize. The icons are the two columns of dots and small squares near the center of the Icon Bar.



## d. Undo

For a demonstration of *Undo*, draw two lines as described in section **b** above, then click the Undo icon at the bottom right corner of the screen. The last line drawn disappears. Another way to Undo in the middle of an operation is to move the cursor to the top of the screen and "out of sight". For example, if you've started a line and you decide that you don't like where it begins, instead of finishing the line and using the Undo option, move the cursor to the top of the screen as if you wanted the end to be under the menu bar. The line draw operation ends and the line disappears.

## e. Erase

Select **Erase Screen** under **EDIT** to completely erase the entire screen.

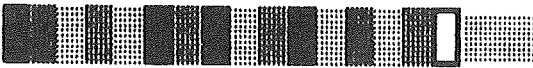
# 2. How To Use Color

## a. Drawing Colors

The current draw color is the color you see in the box to the right of the palette. To make another color the active Draw Color, move the cursor to it and click. The color choices (or palette colors) are shown across the top left side of the Icon Bar.

Experiment by drawing lines, filled circles, and rectangles with various colors.

### Palette colors



#### b. swapping palette colors

Erase the screen as described earlier (from EDIT). Draw a blue line and then a red line anywhere on the screen to use as an example. With the active color red, click on the swap icon (the double-headed arrow next to the active Draw Color square near the center of the Icon Bar - see left). Move the cursor to the blue palette color and click to swap the colors of the two lines. Click the icon again and move the cursor to red to swap back to the original color arrangement



#### c. copying palette colors

Erase the screen and draw another red and blue line. Click blue to make it the active Draw Color. Click the copy icon (the arrow that is pointing to the right next to the active Draw Color square - see above). Move the cursor over the red palette color and click. The blue Draw Color is copied over the red color making both lines blue. Select **Default Colors** under the **COLOR** pull-down to restore your original colors.

#### d. changing a color

The RGB icons are near the center of the Icon Bar. You'll see the letters, RGB, displayed vertically beside 3 rectangles. Each rectangle has a decrease symbol < beside it on the left and an increase symbol > beside it on the right. Lines inside the rectangles show the amount of red, green, and blue in the Draw Color.



### RGB icons

For the example, click on gray near the left end of the palette to make it the active Draw Color. Move the cursor over the increase symbol on the right side of the top RGB rectangle (the one for the red color level). Press and hold down the mouse to increase the red color level to maximum, then release the button. It's at maximum when the line almost fills the rectangle.

Decrease the (G)reen and (B)lue levels to zero by clicking on the decrease symbols on the left side of their rectangles. The original gray color should have changed to bright red. Leave it red for the next section, *blending colors*.

### e. blending colors

Be sure you've done section d above before going on.

Click on the right half of the blend icon (the large icon with an arrow on each end below your color choices).



Move the block part of the blend cursor over the red color you created in section d (near the left end of the palette). Click to blend the entire palette to intermediate shades of red.

Now, click on the characters "+B" at the left side of the blend icon to change the plus sign to a minus sign. This suppresses the color (B)lue in the next blend. Click on the right half of the blend icon and then on the same red to blend it again with the blue color suppressed. You'll see the red blend toward yellow shades this time. Now, click **Undo** and change the "-B" back to "+B" (click it again) to prepare for the next section.

### f. multiple blends

Be sure you've done sections d and e above before going on.

Move the cursor about midway between the left and right side of your color choices (should be all shades of red now) and click on a medium toned red for the active Draw Color.

Use the RGB icons and decrease the red and blue color levels to zero. Set the green color level to about half the maximum level. This should create a dark green color near the center of your palette. Refer back to section d for more on changing the color levels if you need help.

Click on the right half of the blend icon, move the block part of the blend cursor to green, and click to do a blend. Now, your palette should have two separate colors blended.

To see what affect a blend to the left has, click on the left half of the blend icon. Then click on a green palette color. You'll see that blends toward the left get darker rather than lighter.

## 3. How To Load From Disk

Make sure your *Graphics Supermarket* program disk is in the active drive and select **Load** under the **FILE** pull-down. Use the up and down arrows to scroll the list of filenames until you find the file *Space*. Move the cursor over the name, and click twice rapidly (double-click) to select and load it. You can also click once on the name and then on [Open] to load it (or press [Return]).



Once the picture has loaded, click on the up-arrow at the right end of the Menu Bar to see the area hidden by the Menu and Icon Bars.

## 4. How To Save To Disk

You can save the picture loaded above to a new disk by selecting **Save Picture** under **FILE**. First though, you need to put a formatted blank disk (or a disk with enough free space for the picture) in the disk drive. Don't use the *Graphics Supermarket* program disk. There isn't enough room anyway.

After selecting **Save Picture**, type a filename for the picture and press [Return] or click [Save]. Read the *Disk Loads and Saves* section for detailed information.

## 5. How To Change Image Sizes

Choose **Input Frame Size** from the **EDIT** pull-down. Type 50, press [tab], type 50 again, and press [Return]. This sets the Edit Area size (or *Image Size*) to 50 bytes wide by 50 bytes high. Now you should see the Edit Frame on the screen.

To further demonstrate the size change, select **Erase Display Area** from the **EDIT** menu and click [Ok] when you see the confirmation dialog. The area outside the new Edit Area (the Display Area) is erased with the Background Color (black).

Don't forget that your drawing is limited to the area inside the Edit Frame, that is, the Edit Area. There's more about this later.

## 6. How To Use The Clipboard

You should begin with section 3 and work your way to this section since this one uses the screen created by the others. Before you go on, be sure you have some "art work" inside the edit area.

### a. Hold

To save the Edit Area to the Clipboard, select **Hold** under **CLIPBOARD**. Type a temporary name, such as *Sample*, and press [Return] or click [Ok] to complete the hold. Read the help dialog and click [Ok] to continue.

Remember, you can temporarily *Hold* up to 15 images in your computer's memory. To save images permanently to disk, use **Save** under **FILE**.

### b. Paste and Erase

Choose **Paste** from the **CLIPBOARD** menu, read the help dialog, and click [Ok] to continue. Click on *Sample*, or whatever name you used in section a. Move the flashing rectangle to an empty area on any part of the screen and click. Next, click [Cancel] to stop the paste operation.

This is a good time to demonstrate some more erase options. Pick **Erase Edit Area** from the **EDIT** menu and click [Ok] when the confirmation dialog is shown. Notice

that only the area inside the Edit Frame is erased. Now, try drawing a few lines to see how your drawing is limited to the Edit Area inside the Edit Frame.

Return to **EDIT** and select **Erase Display Area**. Click [Ok] when the confirmation dialog is displayed. This time, only the area outside the Edit Frame is erased. Select **EDIT**, then **Erase Screen**, to erase both inside and outside the Edit Frame.

### c. Display - Edit

Select **Display - Edit** under **CLIPBOARD**, read the help dialog message, and click [Ok] to continue. As before, click the name, *Sample*, to erase the entire screen and display the image you held in section a.

### d. Release

Choose **Release** to remove the image, *Sample*, from the Clipboard. After selecting **Release**, click the image's name, and then [Ok] when the confirmation dialog is shown.

### e. Clipboard file loading

Make sure the *Graphics Supermarket* program disk is in the disk drive and select **Load** under **FILE**. Double-click on the filename, *MISC*, to begin the load. After the file loads, read the help dialog and click [Ok] to continue.

Go to **CLIPBOARD**, select **Display - Edit**, and click [Ok]. Select an image and click on its name to display it on the screen with the colors and mode it was created with.

Clipboard images load from disk to the Clipboard and do not disturb or change any image you have on the screen. Full screen pictures load from the disk directly to the screen and replace any images currently on the screen.

## 7. How To Change Modes

Refer to section 6.e above and select any image to display on the screen. Select **640 X 200 Dithered 16 Colors** from the **MODE** pull-down to change to the dithered 640 mode (this mode's colors are created by mixing colors). Choose **640 X 200** under **MODE** to use the solid color 640 mode. And finally, try the **320 X 200 Solid 16 Colors** to return to your original mode. See complete mode descriptions in the *Modes* section.

## 8. How To Print Cards

Several example holiday cards are included on your program disk. They have been clipped down to the actual Image Size that appears on the face of the printed card to conserve disk space, so some preparation is needed before printing them. This section shows how to prepare and print the holiday cards.

### a. loading a card image from disk

Select **Load** under **FILE** (be sure your program diskette is in the disk drive).

Double-click on the filename, *Christmas*, to select and load the card. Click [Ok] when the help dialog is shown.

#### **b. displaying a card**

Choose **Display - Edit** from the **CLIPBOARD** menu. When the help dialog is shown, click [Ok] to continue. Click on the name, *Christmas*, to load the image to the screen.

#### **c. preparing a card for printing**

Now, you need to move the card to the right side of the screen so it will print on the right half of the paper. To do so, go to the **EDIT** pull-down and pick **Input Frame Size**. When the dialog appears, type the number 160 and press [Return] to widen the Edit Area size to the screen's width.

Select **Scroll Parameters** from the **EDIT** pull-down. When the dialog box is shown, type 40 and press [Return]. Now, also under **EDIT**, select **Scroll Edit Area**. Move the arrow to the right side of the screen. Make sure the arrow points toward the right (the direction you want to move the image). Click and wait for the image to scroll. Click a second time to move it completely to the right half of the screen. Finally, move the scroll cursor off the top of the screen (over the Menu Bar) or down over the Icon Bar to stop scrolling and exit the scroll mode. Now the card is ready for printing.

At this point, you could save the prepared card to the Clipboard (with Hold) and then save it to disk for permanent storage. Once saved, you would not have to repeat these steps if you want to print it again. Give the card a different name if you save it in its new form. Refer to the *Disk Loads and Saves* and *The Clipboard* sections if you need help.

If you draw your own cards, you can create them at this size, which is 160 by 166, or you can use full screen size, and draw it on the right half of the screen.

#### **d. printing a card**

Select **Choose Printer** from the **FILE** menu. Click on the name of the printer you have, and the name of the port it's connected to. Make sure the program disk is in the disk drive and click [Ok] to load the printer drivers. *Note: The Control Panel must also be set correctly. If you need help, see the Printing Images section.*

Choose **Print** under **FILE**. Enable the color option if you have an Imagewriter II and a color ribbon. Otherwise, click [Ok] to print the card.

## II. Clipboard

The Clipboard *temporarily* stores images in memory for convenient access while editing. Since Clipboard images are only kept in memory, **be sure you save them to disk** from the **FILE** menu (see *Disk Loads and Saves*) before you quit *Graphics Supermarket*.

The **CLIPBOARD** menu lets you Hold, Display, Paste, Release, Release All, or Animate the images in the Clipboard. The Clipboard holds up to 15 images, and from the **FILE** menu, you can save all the Clipboard images as a single disk file, which is handy for keeping a series of related images in one group. *Programmer's note: Image file structures are explained in the Programming Images section.*

In addition to holding copies of your images, the Clipboard gives you considerable editing power. You can cut and paste images (or parts of them), hold intermediate copies of images while experimenting with changes in shape or color, paste images elsewhere on the screen for reference while editing, or most importantly, you can use the **Hold** menu item and save a copy of your image before performing edit operations that may not be reversible.

The remainder of this section describes each **CLIPBOARD** menu item in detail and shows additional ways you can use the Clipboard for editing. You should also refer to the *Status* section for information on monitoring Clipboard memory usage.

### A. Hold

Select **Hold** to save screen images to the Clipboard. Only the Edit Area, the area inside the Edit Frame, is saved (Edit Frame and Edit Area are explained in the *Introduction*). An image's colors, mode, and size are also saved so you can display and edit the image later.

The Clipboard holds up to 15 images. A "Clipboard Limit Reached!" message is shown if you try to hold more than 15. The following steps show you how to hold an image in the Clipboard:

1. When you have drawn an image you want to hold, select **Hold** from the **CLIPBOARD** pull-down. A dialog box appears and if you have previously held images to the Clipboard, their names appear in a column. Near the top of the dialog box, you'll see an "Enter Name:" prompt with an edit field rectangle beside it. [Ok] and [Cancel] are also shown.
2. Type a name of up to fifteen characters for your image. Characters you type appear in the edit field. *Programmer's note: The image's name becomes its label when source code is generated. See the Programming Images section.*

3. Press [Return] or click [Ok] to complete the **Hold** and exit. You can also cancel the operation and not hold the image by clicking [Cancel].

For a fast Clipboard **Hold**, you don't have to name an image. This lets you speed up a cut and paste operation (described below), or quickly make a backup before performing an irreversible edit operation. To **Hold** without a name, simply leave out step 2 above and press [Return] or click [Ok]. You would normally only do this with images you don't plan to keep, that is, those you plan to release after the operation is completed. Multiple saves without names may cause confusion and are not recommended.

## B. Display - Edit

Select **Display - Edit** under **CLIPBOARD** to view or edit a Clipboard image. You should use *caution* though. Any images on the screen are erased before Clipboard images are displayed (see *Paste* for another alternative), so be sure you hold or save a copy of your screen image first.

After you select **Display - Edit**, simply click the name of the image you want to display. Or you can click [Cancel] to exit and not display.

Images are displayed using the colors and graphic mode they were created with unless you have disabled the *Mode* and *Color* options. If both options are disabled, the screen is not erased when images are displayed, nor are their colors or mode used.

An option is enabled if the box next to it has an [X] inside it, and disabled if it is empty. Click on the boxes to enable or disable the options.

You might want to disable these options if you have pasted other images around the screen for reference and you want to restore an image to a prior condition by displaying an earlier version of it from the Clipboard. With the mode/color options disabled, you could display the earlier version without erasing the screen (or your reference images). Otherwise, you would have to re-paste all your reference images back to the screen.

**Important:** Displaying an image created in the 320 mode into the 640 with the mode option disabled and the color option enabled replaces 640 mode colors with 320 mode colors and is not recommended. The screen display will most likely be affected adversely since color tables in the 640 mode are structured differently than those in the 320 mode (see the *Color Tools* section for more on color tables).

## C. Paste

### 1. Pasting a Clipboard Image

You can use **Paste** to display Clipboard images anywhere on the screen. Pasted

images are transferred from the Clipboard to the screen without changing the current mode or colors, so they take on the colors and mode currently being used. That is, if the current colors and mode are different from those the image was created and held with, the image's colors will be different when it is pasted. You can see an image with it's own colors and mode by using **Display -Edit** with the Mode/Color options enabled.

**Important:** A **Paste** is irreversible once completed so you should save, or hold, current editing before pasting in case the results of pasting are unwanted!

The following steps show you how to paste a Clipboard image to the screen:

- (1) Select **Paste** from the **CLIPBOARD** menu.
- (2) Click on the name of the image you want to paste. The Clipboard display is erased and a rectangle the size of the image you chose is shown.
- (3) Use the mouse to move the rectangle to the place you want to paste the image. Click to complete the paste, or you can cancel by moving the rectangle off the top of the screen.

Once you have pasted an image, the Clipboard display is shown again. You can continue to paste images following steps 2 and 3 above, or click [Cancel] when done.

## 2. Other Uses for Paste

By creatively re-sizing and moving the Edit Frame (see the *Edit Menu* section of *Drawing Tools*), and the Clipboard **Hold** and **Paste** features, you can perform many tasks that go beyond the standard uses for each feature by itself. Some examples are given in this section, but don't hesitate to create uses of your own.

### a. Pasting reference images

Many times when drawing a series of small related images (such as are used in animation), it's quite useful to see the images already drawn while drawing the next in the series. You can use **Paste** to display other images around the screen (outside the Edit Frame) for this purpose, or to see all the images on the screen at once. Of course, the Display Area, (outside the Edit Frame) must be large enough to accommodate the images you want to paste (if you want to edit while viewing other images). There are situations where you might want to paste images inside your Edit Area for editing purposes, but if you want to draw and compare your drawing with other reference images, you wouldn't want the reference images to overlap your drawing.

### b. Cutting and pasting

You can also use **Hold** and **Paste** to perform cut and paste operations. To do a cut, re-size and move the Edit Frame (see *Drawing Tools*) to select the screen area, or image area, you want to cut. Save the area to the Clipboard with **Hold**. Remember, it's the area inside the Edit Frame that's saved to the Clipboard with **Hold**.

After the Clipboard hold is performed, if you want to erase the original image, as you might if you were copying an image to a new location, select **Erase Edit Area** from the **EDIT** menu.

Now, you can use **Paste** to put the image (now in the Clipboard) anywhere you want on the screen. You can release (see the *Release* section) the image from the Clipboard when you no longer need it, and put the Edit Frame back to its original location and size, or change it as desired for editing.

This is also how you can copy a whole or part of an image repeatedly on screen. Just keep selecting the image from the clipboard and paste it wherever you want it on the screen.

### c. Use your imagination

As you can see, in addition to adding power to Clipboard operations, the ability to re-size and move the active Edit Area anywhere on the screen can be combined with many other editing features for even more power. Examples are given in other sections of the manual (for selectively changing pixels/colors and selectively filling - see *Color Tools*). Creative use of the Edit Frame may open doors and solve editing problems unique to your drawing situation. So, *use your imagination*.

## D. Release

You can select **Release** to delete images from the Clipboard. You should use *caution* though. Released images are gone forever unless a copy has been saved to disk (use **Save** under **File**).

To **Release**, or remove, an image from the Clipboard, select **Release**, then click the name of the image you want released. The name you choose is displayed for confirmation. If you are sure you want to release it, click [Ok] and your image is erased from the computer's memory.

**Important:** After a **Release** has been confirmed and completed, it is irreversible! The only way you can get an image back is if you have saved it to disk.

After finishing a **Release**, the Clipboard display is shown again so you can continue to release other images. Click [Cancel] when done.

## E. Release All

**Release All** lets you delete, or empty, the entire Clipboard in one operation.

After selecting **Release All**, a message is displayed asking if you are sure. Click [Ok] to complete the release and empty the Clipboard, or [Cancel] to keep the Clipboard.

**Important:** Once the Clipboard has been emptied, the images cannot be recovered unless you have saved them to disk, so use *caution*!

## F. Animation Test

This feature lets you test animation with Clipboard images. It's especially useful for programmers to test images they plan to animate in a program.

All the images you want to animate should be the same size, and for best results, you should use the same mode that you used to draw the images (i.e. 320 or 640 mode).

The *CREATURE* file on your program disk contains some example images you can use to see how to do animation testing.

To animate the sample images, go to the **FILE** pull-down and load "*Creature.*". Select **Animation Test** under **CLIPBOARD**. Click "Creature2" and "Creature3" to select them for animation. Then click [Ok] to animate. "Creature" is the example used to create the source code samples shown in the *Programming Images* section. Look there for information on using images in your own programs.

After the animation begins, you can click the arrows at the top right screen corner to speed up or slow down the animation. Click [Done] when finished.



# III. Status

The **STATUS** menu gives you information about the current state of the system and editing environment. Look under **STATUS** to check the size of the Clipboard, the size of your current image, or to monitor memory usage.

## A. Byte Width and Byte Height

Look at the **Byte Width** and **Byte Height** items to check the current size, width and height, of the Edit Area. Since images are drawn within the Edit Area, these values also indicate your image size.

You can change the size from the **EDIT** menu or by dragging the Edit Frame to a new size (see *Drawing Tools*). The size may also change when you select **Display - Edit** under the **CLIPBOARD** menu, or when you load images from disk.

## B. Pixel Width and Pixel Height

These two menu items show you the same size information of your edit area as above, but in pixels (the smallest screen dots).

The Edit Area, and your image sizes, are based on byte boundaries as given by **Byte Width** and **Byte Height** above. The screen (or a screen line) is 160 bytes wide in both the 320 and 640 graphic modes. In the 320 mode the screen has 2 pixels per byte ( $2 \times 160 = 320$ ) and in the 640 mode it has 4 pixels per byte ( $4 \times 160 = 640$ ). So the pixel width will always be 2 or 4 times the byte width. Although a byte may hold 2 or 4 pixels in width, it is always equal to 1 pixel in height.

## C. Image Byte Size

You can check **Image Byte Size** to see how much memory your image will use. Remember, a memory location is usually thought of as a byte, and 1024 bytes is 1K (1 Kilobyte).

Your image byte size, plus 60 bytes for other data, is the amount of memory your image will need when you hold it in the Clipboard. So you can use this value along with the other memory information in the **STATUS** menu (see the *Ram Free* and *Largest Block Free* sections below) to monitor memory usage.

## D. Clipboard Byte Size

Look at **Clipboard Byte Size** to see the amount of memory used by all the images you have stored in the Clipboard.

## E. Total Ram Size

The **Total Ram Size** is the amount of memory available in your computer system (Ram = random access memory).

## F. Ram Free

To see how much memory is unused and available, look at **Ram Free**. The amount of free ram decreases when images are saved to the Clipboard (or loaded to it from disk), and increases when images are released from the Clipboard. So if you notice you are running out of memory, or you get a "Can't Allocate Memory!" error message, you may want to save your Clipboard images to disk and release some or all Clipboard images to free up more memory. You will probably need to do this if you have only 512K memory.

There are some limiting factors on how your free memory can be used though. See the *Largest Block Free* section for more on this.

## G. Ram Used

This item shows the amount of your *Total Ram Size* which is currently being used.

## H. Largest Block Free

**Largest Block Free** shows important information for monitoring memory usage. It represents the largest continuous area, or block, of unused memory currently available for use. (See the Introduction I.A.2 for how this may be useful - if you have only 512K memory).

You should only use the **Largest Block Free** value as a guideline for monitoring memory usage since several factors can affect how memory is used. Some of these are described in the next section, *Managing Memory*. Be sure to read this if you are working with limited memory.

## I. Managing Memory

The following information will help you understand some specific situations related to memory usage that could occur while using *Graphics Supermarket* - especially if you have limited memory or push your system to its limit.

To be on the safe side, you should not use up all available memory to ensure the system has enough memory to function properly. *Graphics Supermarket* makes many tool calls which need to allocate and de-allocate memory. No restraints have been placed on how much memory you can use to let 512K systems have as much memory as possible to work with, but it is best to leave at least 3K (more is better) free to ensure the system tools can allocate the memory they need.

Those with more than 512K of memory should have plenty of memory for normal

use, however, 512K users should carefully manage their limited amount of free memory.

Review the following material and examples if you are working with limited memory. However, even though this is only a general discussion of memory usage, it is still somewhat technical by nature. Refer to the Apple IIGS technical manuals for detailed information about memory management.

## 1. Memory Fragmentation

All memory is requested from, and allocated by, the *Memory Manager*. The memory is requested in blocks of various sizes, and is allocated and de-allocated by the Memory Manager as needed. This process can cause memory to become fragmented, leaving areas of free memory scattered between blocks of used memory. The Memory Manager uses memory more efficiently by rearranging blocks where possible, however, fragmentation can still occur (mostly due to restrictions that may be placed on memory when it is allocated -- discussed next).

## 2. Memory Restrictions

When memory is requested from the Memory Manager, it can have restrictions placed on it depending on its intended use. Some possible restrictions on allocated memory are: it may be locked and unmovable; it may have to reside in a given bank; it may not be allowed to cross bank boundaries; or it may need to be page aligned.

Restrictions are only mentioned here so you may see how memory can become fragmented, or why memory may not be allocated even though it appears there is sufficient memory free. For example, blocks which are locked can't be moved, which limits the Memory Manager's ability to compact memory and free larger blocks.

## 3. Clipboard Memory Management

Read on for examples you may encounter while using the Clipboard when your memory is almost full.

### a. Monitoring memory via *Ram Free*

Suppose you determine from checking the **Image Byte Size** menu item that your image will use 10K of memory when saved to the Clipboard (byte size/1024 bytes per K). You look at the **Ram Free** menu item and see you have 80K of free memory - plenty of memory for your 10K image...right?

Possibly not. You could still get a "Can't allocate memory!" error message if **Largest Block Free** shows there is less than 10K due to memory fragmentation.

### b. Monitoring memory via *Largest Block Free*

Now, suppose you still have the 80K of free memory and want to save your 10K image (as above), and you see the **Largest Block Free** menu item shows 11K.

You know you've got it made this time since your 10K image will certainly fit in an

11K memory block, so you do a **Clipboard Hold** and are surprised when you see "Can't Allocate Memory!". This shouldn't happen often but could still occur due to memory restrictions placed on the memory when it's requested from the Memory Manager.

The flip side of this is **Largest Block Free** might show 9K and you may successfully save a 10K image because the Memory Manager was able to compact memory and free a larger block. So, you cannot always predict by monitoring system status whether you will be able to hold your image when the memory size differences are small.

#### 4. If You Run Out of Memory

Remember, you should try not to push memory usage to the limit, but if you must, you should periodically save a copy of important work to disk as a safeguard. Also, in tight memory situations you have nothing to lose by attempting a **Clipboard Hold**. You will be informed if memory isn't available.

If you should run out of memory, you can free memory by releasing Clipboard images that aren't needed, or by saving your Clipboard images to disk and clearing the Clipboard with **Release All**. Your current editing is left untouched.

Another option you could use in an emergency situation is to save your image to disk as a picture file. If your image is smaller than screen-size, go to **EDIT** and select **Full Screen Size** to change its size to the entire screen.

Now you can save the image as a picture file from **FILE**. You can reload it later after you have emptied the Clipboard (or have more available memory). Once reloaded, you can set the Edit Frame back to its original size and re-save your image.

# IV. Graphic Modes

Select the **MODE** pull-down to switch between the super high-resolution graphic modes available. Both the 320 X 200 and the 640 X 200 graphic modes are supported. Throughout this section, and else where in the manual, the 320 X 200 graphic mode is referred to as the 320 mode, and the 640 X 200 mode is referred to as the 640 mode.

You have the option of using 4 solid colors or 16 dithered colors in the 640 mode (see the *Color Tools* section for more information on dithering). The 320 mode always has 16 solid colors.

A mode change occurs immediately when you select **MODE**. The current mode is indicated with a check mark. You can change between the 320 and 640 dithered modes harmlessly at any time. The actual pixels on the screen are not altered, so if you change from the 320 mode to the 640 mode and back, your image is not changed. However, the modes do use different colors, so images appear differently when modes are changed.

You should use *caution* when changing to or from the 4-color 640 mode. As a safe-guard, you may want to save your image to the Clipboard first by selecting **Hold** under **CLIPBOARD** (see *The Clipboard*). That way, you can restore your image if it's adversely affected during the mode change.

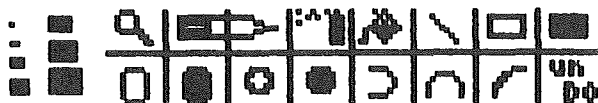
A mode change that involves the 4-color 640 mode can modify the pixels on the screen if the Background Color is out of the range of available colors in the 4 color mode (remember, the other modes have 16 colors). This insures erase operations match the Background Color as they should. If a mode change requires pixel modification, the screen is scanned for pixels which are the Background Color. If any are found, they are changed to a valid color for the 4 color 640 mode. Since the computer doesn't know whether a given pixel is background or part of your image, your image may be modified as well.

# V. Drawing Tools

This section describes the tools you use to draw and edit images. It includes the **EDIT**, **PENSIZE**, and **DRAW** pull-down menus and the icons on the right side of the Icon Bar.

You can select all the menu items in **PENSIZE** and **DRAW** from the Icon Bar. But since you can hide the Icon Bar, all the tools are also selectable from the pull-downs.

The following illustration shows the Icon Bar drawing tools.



## A. Edit Menu

### 1. Undo

**Undo** lets you remove your last draw or edit operation if you don't like the results, or it doesn't turn out as expected. You can select **Undo** from the **EDIT** menu or the Icon Bar. **Undo** is the same regardless of where it is selected.

When you select **Undo**, the screen and colors are restored to the condition they were before the most recent change. You should use **Undo** immediately. Other selections (clicks on icons or the screen area) may disable **Undo** or change what gets undone. **Undo** can only be used once, i.e., you cannot undo the last change and then the one before that, and you cannot undo (or restore) what you've just undone.

### 2. Erase Edit Area

The **Edit Area** is the area inside the **Edit Frame** where you draw and edit your images. To erase the **Edit Area**, select **Erase Edit Area** from the **EDIT** menu.

After making the selection, you are asked, "Are you sure you want to erase current editing?" Click **[Ok]** to erase or **[Cancel]** to keep your current image. Take your time and use *caution* when erasing! After an erase is completed, you can't reverse it!

The **Edit Area** is erased with the **Background Color** (described in the *Color Tools* section). The **Display Area**, or area outside the **Edit Frame**, is not erased.

You can selectively erase parts of the screen by moving and re-sizing the **Edit Frame**. This is described later. You can also select another kind of erase with the mouse from the **DRAW** pull-down or the Icon Bar. (See *Erase* in the **DRAW** section).

### 3. Erase Display Area

This **EDIT** pull-down item erases the area outside your Edit Frame (the Display Area) with the Background Color. The area inside the Edit Frame is not erased.

You can use this feature to erase images you may have pasted on the screen for reference (see *The Clipboard* section).

### 4. Erase Screen

Select **Erase Screen** to erase both the Edit Area and Display Area, in effect, the entire screen. The screen is erased with the Background Color.

After selecting **Erase Screen**, you are asked to confirm the erase. Click [Ok] to erase or [Cancel] to keep the screen as-is. *Caution: After you select [Ok] to confirm the erase, it can't be reversed!*

### 5. About Image Sizes

This is general information about image sizes which may be useful while reading the following sections which describe the **EDIT** pull-down.

*Graphics Supermarket* lets you draw and edit full screens like other drawing applications. However, you also have the option of specifying the size of the image you want to draw. This lets you draw images, or a series of images, of the exact size you need.

You can save your images to disk for use at a later time, combine images to create new ones, cut parts of images and paste them to other images or pictures (see *The Clipboard*), or even generate source code for use in programming (see the *Disk Loads and Saves*, and the *Programming Images* sections). The ability to re-size and move the Edit Area gives you considerable power. The *Clipboard* and *Color Tools* sections show some examples of how you can use this power.

Images can range from 1 X 1 byte to full screen size, which is 160 X 200 bytes. Specifying image sizes with bytes rather than pixels lets the size remain the same regardless of the mode you use. So, you can switch modes and still have the same size image. The size remains the same in width and height values, and in the amount of space it occupies on the screen.

Your image size defaults to the entire screen when you start up *Graphics Supermarket*. You can check the size from the **STATUS** menu (see the *Status* section). Read on to learn to how change the image size, i.e., the size of the Edit Area.

### 6. Drag Frame To Size

If your Edit Frame is small enough to fit on the screen, you can move the Draw Cursor outside one of its 4 corners, press and hold the mouse button, and drag the frame to a new size. This lets you quickly re-size your Edit Area without having to

access the **EDIT** pull-down. You must click *outside* the Edit Frame or the click is interpreted as an edit or draw operation.

If part or all of the Edit Frame is off the screen and you can't move to the outside of the corner you want to use for re-sizing, select **Drag Frame To Size** from the **EDIT** pull-down. Now the program knows you want to re-size the frame so it interprets the next press of the mouse button as a corner selection rather than a draw operation. This lets you select a corner from *inside* the Edit Frame. The corner closest to where you click the cursor is selected for re-sizing.

After you click, hold the mouse button down and drag the frame to a new size, or you can still cancel by dragging the Edit Frame off the top of the screen or down over the Icon Bar. If you do cancel, the frame is set back to its original size.

While you drag the Edit Frame to a new size, you can see its width and height at the top of the screen. You can drag to a specific size or visually re-size the frame as desired. Read on to see how you can input the size you want.

## 7. Input Frame Size

You can also change your image, or Edit Frame size by typing from the keyboard. To do so, select **Input Frame Size** under **EDIT**.

You'll see prompts for entering the width and height. The current size is shown inside the rectangle beside each prompt. Type size changes on the keyboard, then use [tab] to move from one rectangle to another.

Press [Return] or click [Ok] to re-size the Edit Frame. Click [Cancel] to keep the original frame.

The size ranges for the width and height are included in parenthesis with each prompt. The minimum size for each is 1. The maximum size you can have for the width is 160 bytes, which is the screen's width. The maximum size you can have for the height with the Icon Bar visible is 166. If you hide the Icon Bar, you can have a height of 187. The size ranges you can type are the same as those you can click and drag.

If you need a height greater than 187, you can set the size to the entire screen for a height of 200. This is described in the next section.

## 8. Full Screen Size

Choose **Full Screen Size** from the **EDIT** pull-down to set your image size, or Edit Area to the width and height of the screen.

If you set the full screen size, an arrow is shown at the right end of the Menu Bar. Click the arrow to access the screen areas hidden by the Menu and Icon Bars.



## 9. Move Frame

**Move Frame** under the **EDIT** menu lets you move the Edit Frame to a new screen location. After you select **Move Frame**, the Edit Frame begins to flash. Use the mouse to move the frame to a new location. To place it, click the mouse button.

To cancel the move and restore the Edit Frame to its original location, move the arrow off the top of the screen.

Use this feature along with re-sizing to clip (or cut) a portion of the screen for saving to the Clipboard, or to isolate an area for editing. This lets you perform an edit operation (such as pixel swaps, filling, etc.) in an isolated area without affecting your entire image.

Remember, the area *inside* the frame is the active Edit Area, and is the area saved to the Clipboard with **Hold** (see *The Clipboard*).

## 10. Hide Frame - Show Frame

You can select **Hide Frame** to view your image without the Edit Frame surrounding it. When you select **Hide Frame**, the frame is hidden and the menu item changes to **Show Frame** so you can make the frame visible again.

Editing is still limited to the Edit Area even though the frame is hidden. Also, since the entire screen is the same color after a screen erase, the Edit Frame is forced to a visible status during the erase. Showing the frame avoids confusion about the location of the Edit Area. If you want, you can hide the frame again after erasing.

You might want to hide the frame to clearly see your work when you are finished, or if you are drawing near the frame. Another option for the latter situation is to enlarge the Edit Frame and reduce it when editing is finished.

## 11. Horizontal Flip

Select **Horizontal Flip** to flip an image - or Edit Area - from left to right / right to left. You can do a second flip to restore the image to its original orientation.

Use this if you have drawn a figure of a man facing right, for example, and also need one facing left. Use **Horizontal Flip** to get the opposite orientation instantly without having to redraw your image. **Hold** each image to the Clipboard giving each a different name, such as "man.right" and "man.left". You can use **Horizontal Flip** in combination with **Vertical Flip** (see below).

## 12. Vertical Flip

Choose **Vertical Flip** to flip an image - or Edit Area - from top to bottom / bottom to top. A second **Vertical Flip** restores the image to its original orientation.

## 13. Scroll Edit Area

Use **Scroll Edit Area** when you want to move the Edit Area *within* the Edit Frame. A scroll moves the entire Edit Area left, right, up, or down, as you might slide a piece of paper on a table.

However, be careful when scrolling, as any part of your image that scrolls beyond the Edit Frame is discarded. Save your image to the Clipboard or disk before scrolling so if you don't like the results, you can restore your image and try again. You can select **Undo** (immediately after scrolling) to reverse the most recent scroll, unless your image is screen size. A screen size image requires additional memory to scroll, so **Undo** is not available and you should save a backup instead, as mentioned above.

The Following steps describe how to scroll the Edit Area.

- a. Select **Scroll Edit Area** under the **EDIT** menu. You'll see an arrow cursor appear on the screen. The arrow points in the direction your image, or Edit Area, will move when scrolled. Use the mouse to move the arrow around the perimeter of the Edit Area. As you move, the arrow's direction changes according to the side of the Edit Frame (or your image) you are nearest. For example, if you are closer to the left side, the arrow points left, and your image will move left when scrolled. If you are closer to the top, the arrow points up, and your image will move up, and so on.
- b. Move the arrow so that it points in the direction you want to scroll your image.
- c. Click the mouse button quickly to move your image one position. The distance your image moves is determined by the parameters you set under **Scroll Parameters**, which is described in the next section. Press and hold the mouse button for continuous scrolling. Release the button to stop. Continue scrolling until your image is positioned as desired.
- d. When you're finished, move the arrow cursor off the top of the screen.

## 14. Scroll Parameters

**Scroll Parameters** lets you change the scroll speed (increase or decrease the delay time between each scroll), and the distance your image moves when scrolled.

Set vertical and horizontal distances to move your image from 1 to 50 pixels with each scroll. A longer distance is very useful when scrolling a large area horizontally.

Set the scroll delay to a value from 1 to 10. The higher the value, the longer the delay between each scroll. A longer delay gives you more control when scrolling small images. Since small images use less screen area, they scroll quickly. They might

move several positions before you can release the mouse button. So set the delay value high to have more control over the scroll.

Press [tab] to move from one scroll parameter to another and then type the desired value for each. When you've set the parameters, press [return] or click [Ok]. Or you can click [Cancel] to go back to your original scroll parameter values.

## 15. Hide Icons - Show Icons

To remove the Icon Bar at the bottom of the screen, select **Hide Icons** under the **EDIT** menu. The icons are hidden and the menu item, **Hide Icons**, changes to **Show Icons** so you can display the icons again.

You might occasionally want to hide the icons so you can see or access a larger portion of the screen during full screen editing, or you might want to hide them so you can create an image with a height greater than 166 (see *Image Sizes*). *Note: If you set the height higher than 166 but less than full screen size, the icons can't be shown again until the height is below 166 or increased to full screen size.*

## B. Pensize Menu

The Pensize refers to the width and height of the screen area that changes when you draw. For example, if you draw a line, the thickness of the line is determined by the Pensize. The larger the Pensize, the wider the line will be. The Draw Cursor's size indicates the size of the pen you are using.

You can choose a new Pensize under the **PENSIZE** menu or from the Icon Bar. The icon choices are represented by two columns of seven different solid rectangles displayed slightly to the right of the center of the Icon Bar. The choices range in size from a 1 by 1 pixel to many pixels, the number depends on the current graphic mode. A small rectangular frame encloses the currently selected, or active, Pensize on the Icon Bar. The active Pensize pull-down item has a check mark beside it. Menu item pen sizes are shown in pixel widths and heights, and the icon pen sizes are drawn to size. The sizes are designed so each graduation in size appears as square as possible.

To change the Pensize, select the new size from the **PENSIZE** menu or you can click over the desired Pensize icon on the Icon Bar.

## C. Draw Menu

### 1. Magnify

**Magnify** lets you zoom in on an image and see more detail. You can then easily see and change individual pixels.

#### a. Selecting

You can click on the **Magnifying Glass** icon or select **Magnify** from the **DRAW** menu to magnify your image.

Your image is magnified and displayed immediately if it's small enough to magnify all of it. If it's too large to be magnified at once, a selection rectangle is shown. Use the rectangle to select the portion of your image you wish to magnify. *Note: If you need to cancel at this point, use the mouse to move the selection rectangle off the top of the screen.*

To select an area for magnification, position the selection rectangle over the area you want magnified and click the mouse button.

#### **b. Quitting**

After selecting an area for magnification, you'll see the word "Done" displayed on the left side of the Menu Bar. You can click on [Done] to quit and return to normal editing.

#### **c. Magnifying a new area**

If your image is large and you can only select a portion for magnification, a click anywhere in the normal size viewing area below the Menu Bar (described below) lets you select a new area to magnify.

#### **d. Monitoring your progress**

The area you selected for magnifying (and surrounding area) is displayed below the Menu Bar at its normal size with the surrounding area displayed in reverse colors. The selected area is shown with its original colors. This lets you monitor your magnified editing at normal size.

#### **e. Changing magnification levels**

The magnification level, and arrows for changing it, are shown on the right side of the Menu Bar. Click on the [up-arrow] to increase the magnification and zoom in closer to your image, or on the [down-arrow] to decrease the magnification and zoom out. There are 7 levels of magnification.

#### **f. Selecting colors**

The color tools are displayed at the bottom of the screen. Select your Draw Color in the same way as in normal drawing. See the *Color Tools* section.

Some color tools are not selectable during magnified editing. They include Blending, Swapping, and Copying. These tools can only be used during normal editing.

#### **g. Drawing**

Below the normal size display area and above the color tools, you'll see the area you selected for magnification actually magnified. The pixels in the area are enlarged so you can easily select and change their colors.

A small image may not fill the entire magnified area. In this case, drawing is only allowed within the area that represents the image. Remember, just as you cannot draw outside the Edit Area, the same limits apply to magnified drawing.

To draw, move the Draw Cursor anywhere inside the magnified area and click. The

pixel under the cursor is changed to the Draw Color. You can move the mouse while holding the button down for continuous drawing. As you draw, the normal size image below the Menu Bar is also updated for reference.

#### **h. Erasing**

A click on a magnified pixel that is already the Draw Color initiates an erase rather than a draw. Once initiated, you can do continuous erasing by moving the mouse while holding the button down. As always, the Background Color is used for erasing.

## **2. Erase**

In addition to using the mouse to draw, you can also use it to erase. The area being erased is filled with Background Color. See *Pensize* to change the Edit Cursor size for more control over erasing.

To erase, select **Erase** under **DRAW** or click the **Eraser** on the Icon Bar (the eraser end of the pencil).

Move the Draw Cursor to the area you want to erase and click, or press and hold down the button while moving the mouse. A single click erases the area under the Draw Cursor. Holding the button down lets you do continuous erasing.

You can select **Undo** to restore everything you most recently erased.

## **3. Sketch**

This tool lets you sketch, or draw, similar to the way you might with a pencil and paper. Select **Sketch** under **DRAW**, or click the pointed end of the **Pencil** on the Icon Bar.

To sketch, click to start drawing on the area under the cursor, or move the mouse while holding down the button for continuous sketching. The current Draw Color and Pensize are used for sketching. (Refer to *Pensize* and *Color Tools* for more information).

You can select **Undo** to remove everything most recently drawn.

## **4. Spray**

The spray tool draws a spray pattern using the current Draw Color. You may find it useful for texturing, shading, and blending colors.

Select the spray tool from the Icon Bar by clicking the **Spray Can** or by selecting **Spray** under **DRAW**. When you select spray, as you move the Draw Cursor into the Edit Area, it changes to a spray pattern.

To spray, move the mouse while holding down the mouse button. You can lighten and darken the spraying affect by moving faster or slower.

Select **Undo** to remove everything you most recently sprayed.

## 5. Fill

**Fill** lets you color an area inside the Edit Area with the current Draw Color. Select **Fill** from the **DRAW** menu or click the tipped **Paint Can** to select **Fill** on the Icon Bar.

To fill an area, move the Draw Cursor to the area you want filled and click. The color just under the Draw Cursor becomes the *target color* and is filled, or copied over, with the Draw Color.

This **Fill** implementation is vertically limited. The fill spreads left and right on a line until it encounters a color that is not the target color (in both directions). This continues on each line (moving down - then up) until a color that is not the target color is hit directly below and above the start point.

By carefully selecting where you begin the fill, you can control how much of an area is filled. The advantage of this type of fill is the greater degree of control it provides. It is so precise that you can even fill single pixel lines with a new color. However, since it is a vertically controlled fill, you may occasionally need to do more than one fill to color complex areas.

You can select **Undo** from the **EDIT** pull-down or the Icon Bar to remove the most recent fill.

## 6. Line

You can draw lines in the current Draw Color by using the line tool. The Pensize determines the width and height, or size, of a line.

Select **Line**, or click the **Line** icon to activate the line tool.

To draw a line, move the Draw Cursor where you want the line to start. Press and hold down the mouse button, and move the mouse to drag the line to the desired distance and location.

Cancel a line you have already begun by dragging the line off the top of the screen or over the Icon Bar. You can also select **Undo** to remove the last line drawn.

## 7. Rectangles / Ovals / Arcs

The remaining drawing tools are all selected and used in the same manner. These are tools for drawing rectangles, rectangles with rounded corners, ovals, half ovals, and arcs.

As with other drawing tools, you select by clicking an icon or by choosing a tool under the **DRAW** pull-down. The current Draw Color and Pensize are used when drawing.

You can draw rectangles and ovals empty or filled, and you can draw half ovals with a vertical or horizontal orientation. To do so, simply click the appropriate icon, empty

or filled, or with the orientation you want (or select the desired tool from **DRAW**). You can draw arcs with different orientations (between the same two points) by selectively choosing which of the two points you begin the drawing from.

When you select a tool and move the cursor to the Edit Area, *cross hairs* appear so you can easily align rectangles and ovals. The cross hairs are two lines that cross, one vertical, one horizontal. The lines are only visible inside the Edit Area. The point where they meet is the rotation point. You can rotate what you are drawing (a rectangle, oval, etc.) around the rotation point. Try it by selecting a rectangle, for example, and experimenting.

To draw, press and hold the mouse button. Move the mouse to drag your chosen shape to the desired size and orientation. Release the button to complete the draw.

To cancel a draw operation you have already begun, drag off the top of the screen or over the Icon Bar. As with the other drawing tools, you can select **Undo** to remove your most recent draw.





table entry, and the last color on the right from the last (or 16th) color table entry. This has no relevance in selecting colors and drawing, but may help you understand what is taking place when you change colors by changing RGB levels (described later), which changes color table entries.

### 3. The 640 Mode Color Palette

#### a. Using 16 dithered colors

As mentioned earlier, dithered 640 mode colors are created by mixing colors. Even though the dithered 640 mode color table still has 16 entries, each dithered palette color is made from several entries in the color table rather than a single entry as in the 320 mode. So dithered palette colors are interdependent, that is, different colors in the palette share the same color table entry.

This has important implications relative to changing dithered palette colors. Since dithered colors share color table entries, if you alter one color table entry to change a given dithered palette color, you also change every other dithered palette color which shares the same color table entry.

The interdependence of the 640 mode dithered palette colors makes it very difficult for you to effectively change them. When you change one, you may also change others you don't want changed. This could be why other drawing applications have avoided dithered colors altogether. We felt the advantages of dithered colors were worth the risks.

To help you see the relationship between the dithered palette colors and color table entries, a short color bar is drawn over the top of each dithered palette color. The color bar shows the actual color of the corresponding color table entry. If you do experiment with altering a dithered palette color, watch the color bar over it - not the palette color itself. The color bar represents the color table entry, or color, you are actually changing.

You can change palette colors easily and without side effects in the other graphic modes (the 320 and 4-color 640 modes). The dithered 640 mode is the only mode that requires special consideration.

Since dithered palette colors are so difficult to change, you may prefer to change pixels on the screen rather than palette colors when you want to change an image's color (assuming there is an existing dithered palette color you would like to use). You can change colors and get the desired result by copying, swapping, or painting pixels.

The steps for performing these operations are given later in the *Color Icon Tools* section. You can also find detailed descriptions of dithering and color tables in the Apple IIGS technical reference manuals.

#### b. Using 4 solid colors

In the 4 color 640 mode, your Color Palette display contains four colors. All the color tools and features described in the remainder of this section can be used with the 4

available colors. Read on to see how to put the color tools to work for you.

## B. Color Icon Tools

Color icons let you to quickly access color tools. As mentioned earlier, all the color icons are on the left side of the Icon Bar at the bottom of the screen (see the illustration at the beginning of this section). You can hide the icons from the **EDIT** menu (see *Drawing Tools*). If hidden, you select color tools from the **COLOR** pull-down. **COLOR** is explained later in this section. Refer to the illustration shown at the beginning of the *Color Tools* section to locate color icons.

### 1. Draw Color


To select a palette color, move the cursor to the color you want to draw with and click. The *Draw Color* is the palette color you have selected. It is the solid square at the right end of the Color Palette and changes to reflect your selection. All the drawing tools use the current Draw Color.

### 2. Background Color

The *Background Color* is the palette color you have selected as your background. It's usually the color you draw over when you first begin to draw, or the most prevalent color on the screen. When *Graphics Supermarket* first begins, the Background Color defaults to black.

All erase operations erase with the current Background Color. This is important to remember, especially with complex drawings that have many colors. If you forget this, you could become confused in some situations. For example, you might have a drawing where you've filled a large area with blue and the current Background Color is black. You've drawn over the blue with another color, say red. Now you decide to erase the red. You select the erase tool and begin to erase expecting the red to become blue. Instead, it appears you are drawing with black. The erase is working correctly because the Background Color is black. In this case, you would need to draw over the red with blue, or change the Background Color to blue.

#### a. Selecting a Background Color

The current Background Color is shown below the right end of the Color Palette. It's the rectangle that has a smaller rectangle with the Draw Color inside it. The Background Color rectangle is also used as the *Background Select* icon. 

To change the Background Color, move the cursor to the Background Select icon and click. The Icon Bar is replaced with the color icons only. You'll see the word "Background" written on the left side to confirm you are selecting a new Background Color (rather than a Draw Color). You'll also see a square with the words "Update Screen" beside it. This option is explained later.

To select your new Background Color, move the cursor to the palette color you want and click. The current Background Color rectangle and Draw Color square change to reflect your selection. When they are the color you want, click [Ok]. If you decide not to change the Background Color, re-select the original Background Color and click [Ok] to exit. If the Background Color hasn't changed, no background update action is taken.

#### **b. The Update Screen option**

The *Update Screen* option lets you optionally change, or not change, the original Background Color on the screen to your new Background Color selection. An [X] inside the box next to the words "Update Screen" means you want to update the screen, that is, change the screen areas that are the color of the original Background Color to the new Background Color you have selected. If the box is empty, no colors are changed. Simply click on the box to place an [X] inside it, or to remove the [X]. The *Update Screen* option can be very useful in certain situations.

Following are some things to consider about changing the Background Color and using the *Update Screen* option.

- (1) All erasing is done with the current Background Color.
- (2) If you are creating an image and use the *Update Screen* option, any color in the image that is also the Background Color changes to the new Background Color. So you may want to hold a backup of your image to the Clipboard or save to disk first.
- (3) Here is an example where disabling the *Update Screen* option is useful: The current Background Color is black and you load a picture image from disk - a picture you know wasn't created with *Graphics Supermarket*. Pictures saved with *Graphic Supermarket* are reloaded with their original Background Color. In this example the Background Color is unknown. As you look at the picture you've loaded, you see large areas of white and determine white is probably the Background Color used to create the picture.

You may also notice that many images in the picture have black in them. Remember, black is also the current Background Color. What you would like to do is edit the picture, and to do so, it would be desirable to set the Background Color to white. However, if you were to change the Background Color from black to white with the *Update Screen* option enabled, the black in the images would change to white. Since you don't want the images to change, you must disable the *Update Screen* option when you change the Background Color so that the screen colors stay the same. Then, you can edit the picture, and erase operations will erase with the expected color, white.

### 3. Red, Green, and Blue icon tools (RGB)

As mentioned in the Color Palette section, you can change the amount of Red, Green, or Blue - RGB - in palette colors to create your own custom colors. Changing RGB levels is really quite easy, and is explained later in this section. In addition to reading, you can learn a lot by doing, so go ahead and experiment.

Some of the material in this section is somewhat technical. It's included so you can see some unique problems you can encounter with dithered palette colors. If you have not read the Color Palette section, you should read it before reading this section, and especially before changing palette colors in the dithered 640 mode.

#### a. About RGB levels

The amount of the colors - red, green, or blue - in each palette color can vary from 0 to 15 for a total of 16 levels. This lets you create custom palette colors from a total of 4096 possible variations (16 X 16 X 16 - or 16 levels for each - red, green, and blue).

When you change RGB levels in a palette color, the color table entry (a numerical value) for the palette color is changed. It's the color table entry that determines the color of a palette color. This is especially important when you work with dithered palette colors in the 640 mode. As discussed in the Color Palette section, dithered palette colors are created from several color table entries, not just a single entry as in the 320 mode. And different dithered palette colors share many of the same color table entries.

So, when you change a dithered palette color's RGB levels, you change a single color table entry. Not only will the selected color NOT change as expected, other dithered palette colors that share the same color table entry will change color too.

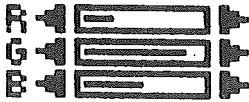
You can see the color you are actually changing (as determined by the color table entry) by looking at the short color bars at the top of the palette colors. Each color bar shows the true color that corresponds to each color table entry. Watch the color bar change as you experiment with changing RGB levels in a dithered color. The bar shows the true color you are actually changing. Read the *Color Palette* section for more on this.

As you may have concluded by now, it's difficult to effectively change, copy, swap or blend palette colors in the dithered 640 mode. When you experiment with changing palette colors, if you don't like the results, you can restore the default colors by selecting **Default Colors** under **COLOR**. It is easier to change palette colors in the 320 or 4-color 640 mode without these unusual side affects. **Undo** can also be used to reverse palette color changes.

But wait! You do have another option for changing image colors in the dithered 640 mode. You can paint, copy, or swap *pixels* on the screen instead. These operations have the expected results when used with dithered screen colors, and are explained later in this section.

## b. How to change red, green, and blue levels

The location of the icons for changing red, green, and blue (RGB) levels in palette colors is illustrated below.



To change the RGB levels, select a current Draw Color (Move the cursor to a palette color and click the mouse).

The letters, **R**, **G**, or **B**, are shown on the left side of each icon. The icon with the letter **R**, for example, is used to change the amount of red in the Draw Color.

Each icon is a rectangle with a triangle on each end. Click the mouse on the right triangle to increase the amount of red (e.g.) in the Draw Color or on the left to decrease the amount of red. A line inside each rectangle shows the current RGB levels in the Draw Color (the longer the line, the higher the level). In the dithered 640 mode, the lines show the amount of color in the color table entry that corresponds to the current Draw Color - that is the amount of color in the thin color bar above the palette color.

To change the RGB level, press and hold down the mouse button while watching the Draw Color square and the screen image change. This is the reason this feature was purposely placed within easy reach on the Icon Bar. Simply click the color you wish to change, click it to make it the current Draw Color and change the RGB levels by clicking one of the triangles.

## 4. Swapping and Copying Colors

This section describes how to change your image's colors by swapping or copying palette colors, or by swapping and copying colors right on the screen.

Beginning from the top left side of the Icon Bar and moving right, are the palette colors, the current Draw Color square, and beside it, the swap and copy icons.



The swap icon is the double-headed arrow, and the copy icon is the arrow that points to the right.

Below the Draw Color square, is either the word "colors" or "pixels" with an up-arrow beside it. When you swap or copy with "colors" visible, palette colors are swapped or copied. If the visible word is "pixels", colors on the screen swap or copy, but the palette colors do not change. The "pixel"/"color" indicator shows whether the change will be just on-screen (and permanent in the case of full screen images) or on

screen and in the palette (not permanent). Each way has its advantages. They are described later in this section. To change from "colors" to "pixels", click the [up-arrow] beside the word.

### **a. Swapping colors**

Select **Swap** to exchange two colors, where every occurrence of one color is replaced by another and vice versa. The swap icon is the double-headed arrow next to the current Draw Color square. One side of the arrow points to the Draw Color, and until swap is selected, the other side of the arrow points to an empty box. The procedure for selecting the colors to swap is described a little later.

As discussed earlier, you can set the word below the icons to "colors" or "pixels" to select the type of swap.

If you swap palette colors, each palette color takes the place of the other in the Palette itself, and the screen colors also switch. You can undo this type of change.

To swap screen colors only, set the swap-type word to "pixels"; the screen is scanned and the colors reverse, but the Color Palette does not change. If it's a full screen, this is permanent and can't be undone.

To do a swap:

- (1) Click a palette color to set the Draw Color to one of the colors you want to swap.
- (2) Click the double-headed arrow to select **Swap**. The arrow cursor changes shape indicating you are in **Swap** mode.
- (3) Move the cursor over the palette colors. As you move, the color you are over fills the box to the right of the double-headed swap arrow. When you see the color you want in the box, click the mouse and the colors swap on the screen if "pixels" is selected, or on screen AND in the palette if "color" is selected.

When a change might cause adverse effects to the screen display, a confirmation message is displayed. Click [Ok] to confirm the swap, or click [Cancel] to stop it. You can also click anywhere outside of the Color Palette to cancel - when the swap box is still empty.

**Note:** There are two different confirmation messages for the two swap types, "colors" or "pixels" -- as "pixel" changes may be permanent, while "color" changes are not.

### **b. Copying colors**

Select **Copy** to copy one color over another. The copy icon is the bottom arrow that points right, from the current Draw Color square to the empty box. The Draw Color will copy over, or replace, the color you put in the box. The procedure for selecting the colors is described later in this section.

As with a swap, you can set the word below the icons to "colors" or "pixels". Click the [up-arrow] beside the word to change it.

When you copy with "colors", the Draw Color copies over the other leaving you with two equal palette colors. Screen colors that were the second color, become the Draw Color. This can be undone.

When you copy screen colors with "pixels", the screen is scanned and the color copied over changes to the Draw Color, but the Color Palette does not change. This cannot be undone with full-screen images.

To do a copy.

- (1) Click on a palette color to set the Draw Color. This color will replace the other color you select (see 3, below).
- (2) Click the bottom arrow to select **Copy**. The arrow cursor changes shape to indicate you are in **Copy** mode.
- (3) Move the cursor over the palette colors. As you move, the colors you pass over fill the box to the right of the copy arrow. The color that fills the box is the **target color**, and it will be replaced by the Draw Color when you complete the copy. Set the target color that you want replaced with the Draw Color and click the mouse button to copy.

A confirmation message displays when the color change might cause adverse affects to the screen display. Click [Ok] to confirm the copy, or click [Cancel] to stop. You can also click anywhere outside the Color Palette to cancel - when the copy box is still empty.

### c. Changing palette colors verses screen pixels

The previous sections show how to swap, copy colors in the Color Palette and the screen, or on the screen only. This section discusses the advantages and limitations of each.

An advantage to changing palette "colors" is that you can more easily reverse the change if you don't like the results. After you use a color tool to change the Color Palette, you can click **Undo** to restore your original colors. You can't always reverse color changes done on the screen when you use the "pixels" option.

For example, draw two balls on the screen, one green, and one red. To make both balls red, there are two copy options.

- (1) Set the copy-type to "colors" and copy the red over the green, or
- (2) Set the copy-type to "pixels" and change the green ball's pixels to red. *Note: You can also paint (or fill) pixels by selecting the [Paint Can] and filling the green ball with red, but that doesn't satisfy the*

*purpose of this example. See the Drawing Tools section for more on filling.*

If you copy palette colors with the "colors" option on, you can click **Undo** and easily restore the original colors, or even manually change the green ball's palette color back to green (by changing RGB levels). However, if you want both balls red, you must keep two reds in the palette. With 16 drawing colors in the 320 mode, this isn't important unless you do it several times. Another possible disadvantage is that every occurrence of a color on the screen changes when you change the palette color. If the entire screen is green, you can't change half of it to red. Read on to see how to change colors on parts of the screen.

The second option ("pixels") doesn't change the palette, so green is still available, but changing colors on the screen has its limits. Suppose you want to restore the green ball. You can't copy green back over the red ball because both balls are red now, so both would be changed to green.

This is a simplified example for the purpose of showing the basic differences between the two options. There are actually many ways you can change colors, and **Undo** is available for "pixel" changes if your image is not screen size.

This paragraph describes how to use the Edit Frame to edit or color a portion of the screen. All operations like swapping, copying, painting, etc., take place in the Edit Area only (within the Edit Frame). Referring to option 2 above, you could move and size the Edit Frame so only the ball you want to change is within the Edit Area. Then you can copy green over the ball that was changed to red without affecting the other red ball. Use the Edit Area (or Edit Frame) to selectively change colors or edit part of your image, leaving areas outside the frame unchanged.

Since it's difficult to change dithered palette colors (see *Color Palette*), you'll want to change colors with the "pixels" option when drawing in the dithered 640 mode.

Always play it safe and save a backup to the Clipboard or disk (or both) before you try any of these edit operations.

#### **d. The effect of color changes on menus**

The screen display is affected when you change black and white palette colors (color table entries). Menus and messages are displayed in black and white. In the dithered 640 mode, there are several color table entries that are black and white (look at the short, thin color bars over the palette colors). Modifying any of them may affect the display, so we suggest that you change pixels rather than palette colors in the dithered mode to avoid adverse screen-display changes.

If you anticipate creating an image with large amounts of black or white, and you think you may want to change either one to another color later, you could make two other palette colors black and white and use those to draw your image. Then if you



need to change one later, you can do it without changing the black and white screen display.

## 5. Blending Colors

To create gradient shades of palette colors with the *blend* tool, use the blend icon below the Color Palette. It's a series of vertical bars with an arrow at each end.



### a. Using and inhibiting red, green, and blue

The letters "RGB" preceded by either a plus or minus are left of the blend icon. They refer to the colors red, green, and blue. A plus indicates the color is used in a blend. A minus inhibits its use. Click the letter to change between plus and minus.

You can enable and disable the use of red, green, and blue, to blend toward different colors. If all the colors are enabled (all 3 letters are preceded by pluses), a blend toward the right creates gradient shades between the selected color and white. A blend toward the left creates shades between the selected color and black. You can set the pluses or minuses to inhibit or use red, green, and blue; or to blend toward other colors, such as blending blue toward red, or blue toward green. The *How to Try Out Graphics Supermarket* section in the *Introduction* shows how to create multiple color palette blends.

### b. Selecting the blend tool

Click on the right half of the blend icon if you want to blend toward the right (and lighter), or click on the left half if you want to blend toward the left (and darker). The cursor changes to a small block with an arrow beside it. The arrow points in the direction you chose to blend. To blend, put the block cursor over the color you want to blend and click the mouse button. The palette, from the block to the right (or left) immediately changes, as does the screen

You can click **Undo** immediately after a blend to restore your original colors, or select **Default Colors** from the **Color** pull-down. *NOTE: Default color initializes the palette to the original standard colors, while Undo restores the palette to the colors it had before the blend (which may have been custom colors, not default colors). To cancel, click anywhere outside of the Color Palette.*

# VII. Disk Loads And Saves

This section describes the **FILE** menu disk operations. It shows how you can load images from disk, save images to disk, and delete files on a disk.

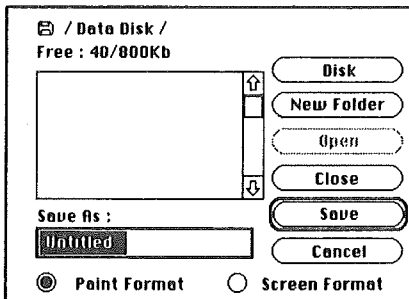
Before you select a **FILE** pull-down option, insert a disk in the active drive. The disk is scanned and the names of the files are listed in a dialog box. Load and Save dialog boxes are different. Each **FILE** option and the dialog boxes are explained here.

## A. Disk Saves

*Graphics Supermarket* lets you save images to disk in several formats, depending on your needs.

### 1. Save Dialog

The following dialog box appears when you select **SAVE**.



When saving to disk, the files shown in the save dialog are dim. Dimmed files cannot be selected, but you can scroll through the list to see all the files.

To save, type a filename of 12 characters or less. The name appears inside the Edit Field at the bottom left of the dialog box. Click [Save] or press [Return] to save.

Following is a description of each save dialog button (click a button to select it) and the types of saves you can do.

#### a. Cancel

Select **Cancel** to return to editing if you decide you don't want to save.

#### b. New Folder

Type a name (as described earlier) and click **New Folder** to create a subdirectory.

### c. Close

**Close** is dim and not selectable unless you have already cataloged a subdirectory (see *Open*). In that case, select **Close** to return to the next highest directory level.

### d. Disk

If you are using more than one disk drive, click **Disk** to switch drives and catalog (or save to) a disk in another drive. When you click **Disk**, the computer looks for disks in your other drives. You may have to click **Disk** more than once to get the drive and disk you want.

### e. Open

**Open** is dim and not selectable unless the current directory has one or more subdirectories. If it is not dim, you can click **Open** to catalog or save to subdirectories.

## 2. Picture Saves

Use **Save Picture** under **FILE** to save screen size images, or pictures. The image size must be full-screen to use **Save Picture**. See the *Drawing Tools* and *Status* sections for more on image sizes.

**Save Picture** saves the screen as a standard Apple IIGS picture with a file type of \$C1. Use **Save Picture Binary** to save the screen as a binary file. Use the binary save to convert pictures to other operating systems, such as DOS 3.3. Use **Save Picture** unless you specifically need **Save Picture Binary**.

## 3. Clipboard Saves

Under **FILE**, the Clipboard save items let you save one or all Clipboard images. Select **Clipboard Save One** to save a single image, or select **Clipboard Save All** to save all the images in one disk file. If you select to save one, after the "binary/source" dialog box appears (see below) the Clipboard dialog box displays a choice of images you can save. Click the image's name to save the image to disk.

After you select a clipboard save, an option dialog box appears where you can choose to save Clipboard images in Binary or as Source Code. Click the button beside the desired option, then click [Ok] to continue, or [Cancel] to cancel.

A solid (or filled) button indicates an option is selected. Clipboard saves default to save Clipboard images in binary unless you need Source Code. Source Code files load and save more slowly, are much larger, and therefore use considerably more disk space than binary files. If you are a programmer, choose the source code option to convert and save images to source code files.

Don't save large images as source code; this would result in gigantic files. Images covering a third of the screen (or even less) generate a lot of source code. Note that you can also use binary images for programs. The *Programming Images* section

gives the information for both options, and describes the additional options you can choose for source code saves.

*Note: When you save images as source code, the file has a ".GS" extension appended to it. The extension separates source image files from other source files when accessing the disk.*

## B. Disk Loads

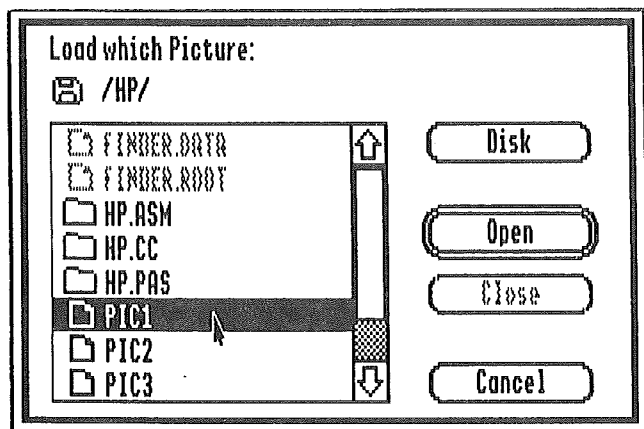
Use **Load** under **FILE** to load picture and Clipboard images from disk. After selecting **Load**, a dialog box shows the names of the files on the disk.

To load a file (can't be dimmed - explained later), click its name to highlight it, then click **Open** or double-click the name to select and load a file in one operation.

A description of the load dialog box and the images you can load follows.

### 1. Load Dialog

When you select **Load** under **FILE**, or **Delete File**, the following dialog box appears.



The **Load Dialog** shows the files on the currently active disk. The dim file names do not have the proper format for *Graphics Supermarket* (more on this later). Click the arrows or the slide bar to scroll the list of file names.

The following items describe the dialog box options.

#### a. Disk

If there is more than 1 disk drive, you can click **Disk** to switch drives and catalog, or load from, a disk in another drive. Do it more than once if you do not access the

## b. Open

**Open** lets you load the currently selected (highlighted) file. Click a filename to select, or highlight, it. Double-click a filename to select and load it in one operation. Click *Open* to select and list a subdirectory, then double-click a filename to load.

## c. Close

**Close** is dim and not selectable until you catalog a subdirectory (see *Open* above). After you do this, you can click **Close** to catalog the next highest level directory.

## 2. Picture Loads

*Graphics Supermarket* loads standard Apples IIGS picture files with a file type of \$C1. Binary files that might be graphic images are also selectable to provide maximum flexibility for loading images that weren't created with *Graphics Supermarket*.

Some selectable binary files may not be graphic images. If an invalid format is discovered after a picture load, the error shows on the screen. However, any loaded data is displayed to provide every opportunity to load a nonstandard image. If the data is not useful, you can load another picture or erase the screen. Binary clipboard file data is discarded if the format is incorrect.

When you load a picture saved by *Graphics Supermarket*, its graphic mode, Color Palette, and Background Color are restored. If you load a standard picture saved by another drawing program, it uses *Graphic Supermarket*'s current mode, Background Color, and the picture's Color Palette. If the colors don't look correct, you can choose **Default Colors** under **COLOR** to restore standard colors.

**Important:** A picture loaded to the screen will replace current editing. If you have an image on the screen, be sure to save it to the Clipboard or disk before you load a picture or you will lose your the current editing.

## 3. Clipboard Loads

When you load from disk, Clipboard files are recognized and loaded automatically to the Clipboard. They can contain 1 to 15 images, and may be binary or source code files. A description of each file type follows. The *Disk Saves* section has more on Clipboard files.

### a. Binary Clipboard loads

*Graphics Supermarket* loads binary Clipboard images from disk directly to the Clipboard. A message displays after the load to inform you that images were loaded to the Clipboard. Use the **CLIPBOARD** menu items to view or edit the images (see *The Clipboard* section). Since the images are loaded to the Clipboard and not to the screen, any current editing is left intact.

### b. Source code Clipboard loads

You can load assembly language and C source code images previously saved by

*Graphics Supermarket.* Image source files are distinguished from other assembly language and C source files with a ".GS" extension. *Graphics Supermarket* appends the extension to source files when it saves them.

Source code images are loaded to the Clipboard just like Binary loads (see 3 above). You can display or edit them from the **CLIPBOARD** menu.

**Important:** Don't change the structure of source files (like using an editor and deleting lines) or they won't load properly; the load cancels and a file structure error is reported. You can change a source filename but it must end with a ".GS" extension, or you can't select it.

*Programmer's note: Use source files "as is," in case you need to edit them later, or save them as binary for later editing and create the source code you need for programming only; copy the source into your program and change it as desired while keeping an original version (source or binary) for editing. See [Programming Images] for more.*

## C. Deleting Files

Select **Delete File** under **FILE** to permanently remove a file from disk. After selecting **Delete File**, the disk in the active drive is scanned and the dialog box displays a list of the file names.

The dialog box is identical to the one used for loading files, except all file names are selectable (see *Disk Loads* for a description of the dialog box). Double-click a filename to select and open it for deleting. You can also click a name once to select (and highlight) it, then click **Open** to open it for deleting.

After you select and open a file to delete, a confirmation dialog appears. If the filename is correct, click [Ok] to permanently delete the file from the disk. Click [Cancel] to keep the file and return to editing.

# VIII. Printing Images

## A. About Printing

### 1. Menu Items For Printing

**Choose Printer**, **Page Setup**, and **Print** under **FILE** help you set up your system to print images. The next three sections briefly describe each menu item, and refer you to more detailed information.

#### a. Choose Printer

Before doing any printing, select **Choose Printer** to tell *Graphics Supermarket* about your printer, and to which port it is connected. (The *Print Manager* only supports the standard serial ports at this time - not plug in cards). You must use **Choose Printer** once each editing session, but only if you want to print.

Use the *HGS Control Panel* to tell the computer which port is used. The *Control Panel* section (A.3) helps set up the port, and the *Choose Printer* section (B) shows how to choose your printer type.

#### b. Page Setup

Select **Page Setup** to set the paper size and the direction to print. This is optional. If **Page Setup** isn't used, the computer reverts to default settings. See *Page Setup* section (C) for more detailed information.

#### c. Print

Select **Print** to set the number of copies, whether to print in color, and to actually print. Before printing, be sure you have followed step 1 above, or your computer may hang-up - forcing you to re-boot. Save your images to disk as a safeguard before you print. See *Print* section (D) for a complete description of **Print** options.

### 2. Memory Requirements For Printing

If you have a 512K system, or you don't have enough free memory for other reasons (such as ram disk allocation or clipboard size), you can not use the standard dialog box options for printing. When you select **Page Setup**, a message may say you need more memory. When you select **Print**, you see a dialog with a color option. Set the color option appropriately and click [Ok] to continue or click [Cancel] to return to editing. *Note: You still must use [Choose Printer] to select a printer and port.*

If there is more than 512K available (that's not being used by a RAM disk, etc.), some images can be saved to disk, and **Release** can be used to free enough memory to use the standard options. See the *Clipboard* and *Status* sections for more on releasing images and memory management.

### 3. Control Panel

This information helps set the Control Panel for printing. Refer to your IIGS owner's manuals for details on the Control Panel.

First, select **Choose Printer** under **FILE**. Choose the printer and its port. (See *Choose Printer* in Section B for selecting). Follow the directions below to access the Control Panel.

- (1) Press the [apple-control-esc] keys simultaneously to access the *Desk Accessories* menu.
- (2) Use the [up/dn arrow] keys to highlight **Control Panel** and press [Return] to get to the *Control Panel* menu.
- (3) Use the arrows to highlight **Slots** and press [Return].
- (4) Now, use the arrows to highlight either **Slot 1** if your printer is plugged into the printer port, **Slot 2** if your printer is connected to the modem port, or **Slot 1** if your printer is an *Apple Talk* set up.
- (5) Use the [right-arrow] to change the highlighted slot to the appropriate setting as follows:

If the port you chose was:

- a. Printer: Set **Slot 1 to Printer Port**
  - b. Modem: Set **Slot 2 to Modem Port**
  - c. Apple Talk: Set **Slot 1 to Your card** then highlight and set **Slot 7 to Built-in Apple Talk**
- (6) Press [Return] to keep the new settings, highlight **Quit**, and press [Return] again to return to the *Choose Printer* dialog box.
  - (7) Make sure your program disk is in the active drive and click [Ok].
  - (8) Now save all your images to disk, and then turn the computer off and, after 15 - 30 seconds, on to initialize the system with the new Control Panel settings. Re-boot *Graphics Supermarket* and load the images you want to print. Refer to other parts of this section if you need more help.

## B. Choose Printer

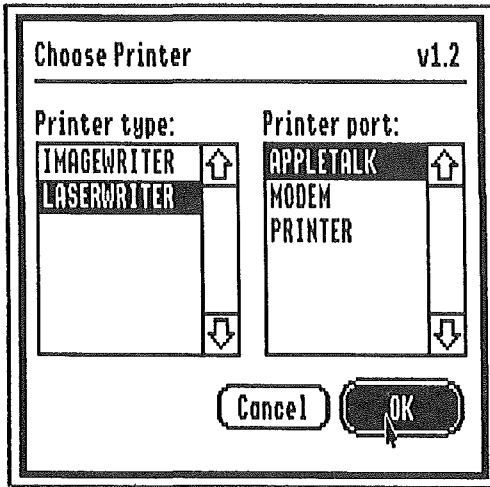
**Choose Printer** under **FILE** lets you select your printer, and the port it's in.

This dialog box appears after you select **Choose Printer**.

(See next page.)



## Choose dialog



The **Choose** dialog lists the printers and ports you can select. Click to make a selection and highlight a name. If necessary, you can scroll the list by clicking the scroll bar arrows, or by sliding the bar.

Select your printer from the *Printer Type* list, and select its port, (where its connected on the back of the computer) from the *Printer Port* list. *Note: You must also have the Control Panel slots set correctly or your computer may hang-up when you try to print. You can do it from the Choose dialog by pressing the [apple-control-esc] keys simultaneously. To see how, refer to A.3 above.*

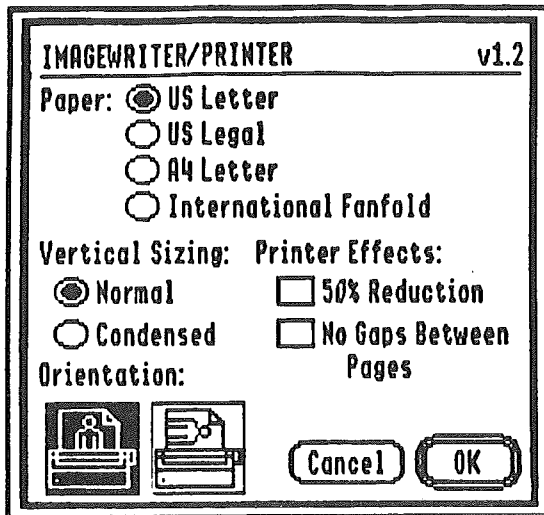
Once you've selected the correct printer and port, make sure your *Graphics Supermarket* program disk is in the active drive, and click [Ok]. The printer and port drivers load and install automatically. If you don't want to load new drivers, you can click [Cancel] to restore your original settings and exit from **Choose Printer**.

## C. Page Setup

To select paper size, and the way you want your image to print on the page (its orientation), select **Page Setup**. The dialog box which appears is different for a Laserwriter than an Imagewriter. This is the dialog you'll see for an Imagewriter.

*Note: If you have a 512K system, or you don't have enough free memory, you can not use the standard dialog options. Instead, a message appears that says you need more memory. (See the Memory Requirements For Printing section.)*

## Imagewriter Page Setup dialog



To select **Paper**, **Vertical Sizing**, or **Printer Effects** click the button or box beside the desired option. When selected, a button is filled, and a box has an [X] inside it.

You can print with an upright *Portrait* or a sideways *Landscape*. Click an orientation icon to select it. Following is a description of these and other **Page Setup** dialog options.

### 1. Paper

#### a. US Letter

This is the default paper setting for 8-1/2 by 11 inch paper.

#### b. US Legal

Use this option for paper that's 8-1/2 by 14 inches.

#### c. A4 Letter

This option is for European standard paper. It's size is 8-1/4 by 11-2/3 inches (210 by 297 mm).

#### d. International Fanfold

This option is for the Imagewriter only. It's the International standard size of 8-1/4 by 12 inches (210 mm by 12 inches).

#### e. B5 Letter

This option is for the Laserwriter only (see the next dialog illus.). Its paper size is 15 by 11 inches (176 by 250 mm).

## 2. Vertical Sizing

### a. Normal

This setting prints images with the same proportions they have on the screen.

### b. Intermediate

*Intermediate* prints images approximately two thirds their normal size (for the Laserwriter only).

### c. Condensed

Images printed with this option are about half their normal height.

## 3. Printer Effects

### a. 50% Reduction

Images are printed at half their normal width and height.

### b. No Gaps Between Pages

Printing is continuous without page breaks when this option is selected.

## 4. Orientation

### a. Portrait

Images are printed with the same upright orientation they have on the screen.

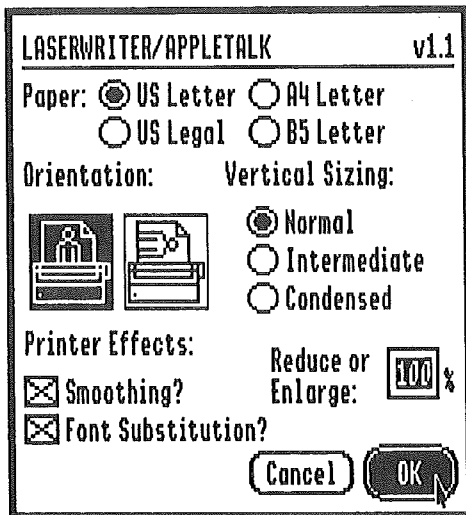
### b. Landscape

Select this option to print images sideways (they are turned right 90 degrees).

The Laserwriter **Page Setup** dialog box is different than the Imagewriter's. Only the options that are different are described below. All other options are the same as the Imagewriter's. Here is the **Page Setup** dialog box for the Laserwriter.

(See next page.)

## Laserwriter Page Setup dialog



### 5. Printer Effects

These Laserwriter options are for bit mapped fonts and are ignored.

### 6. Reduce or Enlarge

You can type a number from 25 to 200 to reduce or enlarge your image print-out. 100 is normal size (100% of image), 200 enlarges an image to twice its size (100% X 2 = 200%), and 50 reduces an image to half its normal size (100% / 2 = 50%).

## D. Print

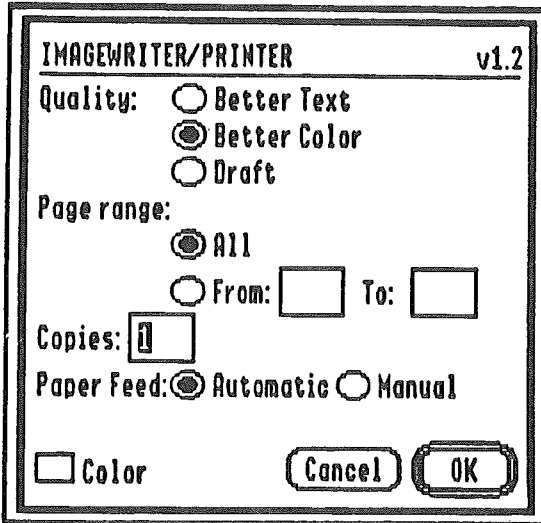
After you have chosen your printer with **Choose Printer** under **FILE**, and you've used **Page Setup** to choose the paper size and orientation you want (optional - the default settings are used if you don't change them), you can select **Print** to print your image.

*Note: If you have a 512K system, or there is not enough free memory, you can not use the standard dialog options. Instead, you'll see a dialog with a color option. Set the color option appropriately and click [Ok] to continue or [Cancel] to return to editing. See the *Memory Requirements For Printing* section for more.*

Here is the **Print** dialog box for an Imagewriter, and a description of its options. The Laserwriter dialog box is shown later.

(See next page.)

## Imagewriter Print dialog



### 1. Quality

#### a. Better Text

This option is used for high quality text and ignored.

#### b. Better Color

This option is always used with an Imagewriter II color ribbon. It produces the highest quality print-out.

#### c. Draft

*Graphics Supermarket* ignores a *Draft* selection. Its mostly used for dumping text to the printer.

### 2. Page Range

This is used only for text and is ignored by *Graphics Supermarket*. The image inside the Edit Area is printed and sets the range.

### 3. Copies

Type the number of copies you want to print. The paper automatically moves to the top of the next page and prints another copy (if the *Paper Feed* is set to Automatic -- see below).

## 4. Paper Feed

### a. Automatic

Used for continuous paper. The paper advances and continues to print multiple copies.

### b. Manual

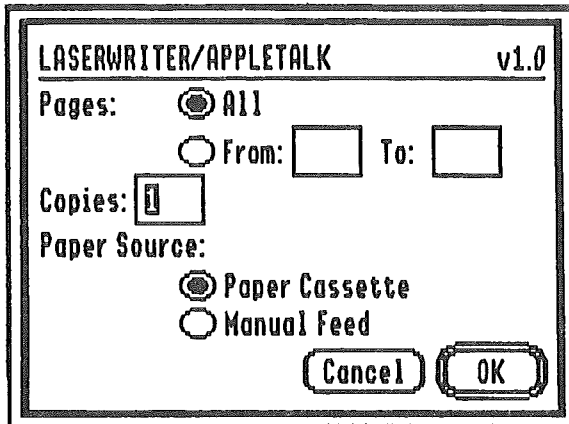
For single sheet paper, if printing multiple copies, you are prompted before each copy, and can continue printing or stop.

## 5. Color

If you have an Imagewriter II and a color ribbon, select this option to print images in color. An Imagewriter I can't print in color, so don't use this.

Here is the **Print dialog** for the Laserwriter.

### Laserwriter Print dialog



## 6. Pages

As previously discussed in section 2, this option is not used.

## 7. Copies

Type the number of copies you want to print. Use [Tab] to move to the "copies" edit field rectangle for typing.

## 8. Paper Source

### a. Paper Tray

You can use this option to select a paper tray.

## b. Manual Feed

Use this option for feeding single sheets of paper by hand.

## 9. To Start Printing

Once you have the **Print** dialog options set as desired, click [Ok] to start printing. Make sure your printer is on and ready. Press the [apple-period] keys simultaneously to stop printing in progress (if you are using standard dialogs and options; a dump performed when you don't have enough memory for standard dialogs won't stop the printing).

Be sure the IIGS Control Panel is set correctly for your printer, or everything may hang-up when you try to print. (See the beginning of this section for more on the Control Panel.)

If you decide not to print, click [Cancel] to leave **Print** and return to editing.

Note: Use white as a transparent color. White areas are not printed. Use a white background when you want to print only the foreground colors. (See *Color Tools* for how to change the background to white, or how to fill areas with white). You can also invert the Edit Area to print a negative of your image which might be useful if the background is black (it will change to white). The Introduction explains how to print cards.

# IX. Programming Images

This section shows how to use images in your own programs. Section A describes how to save Clipboard images as C or assembly language source code. Section B provides information on Clipboard binary image files, and section C gives a complete description of all the file formats.

Section C, *File Formats*, not only describes the structure of source and binary Clipboard files, it also provides several programming examples on how to display *Graphics Supermarket* images to the screen. There are examples for C, assembly language, and Applesoft BASIC (running under Prodos or Dos 3.3).

## A. Generating Source Image Files

*Graphics Supermarket* lets you save images as assembly language or C source code. You can draw images and quickly convert them to source code. No more plotting and converting images to source with graph paper, nor trying to figure out ingenious ways to use images created with paint programs not designed with the programmer in mind.

Source saves are intended for relatively small images, such as animated characters. You can save large images more efficiently in binary. See the *Binary File Format* section for more.

Since the Apple IIGS uses 32000 bytes for a screen display, large images would generate huge amounts of source code. In fact, you can create source code files so large some editors may not load them, especially if you save several large Clipboard images in the same file. If a file is too large, you can decrease its size by breaking it into more than one file, with fewer images in each file.

The process of converting binary to ASCII for source saves, and ASCII to binary for source loads takes time, so be prepared for slower loads and saves.

The remainder of this section shows you how to save images as source code, and briefly describes source code save options. More detailed information on source file structures is in *Source File Format*.

### 1. How To Save Source Code

Images you want to save as (or convert to) source code must be in the Clipboard. You can save a single image or all the Clipboard images at once.

Go to **FILE** and select either **Save Clipboard One** or **Save Clipboard All**.

When the *Binary / Source* option dialog appears, click the button beside *Source*, then click [Ok] to continue.



If you select **Save Clipboard All**, an option dialog box lets you set the following options.

#### **a. Save all options**

##### **(1) All with color table**

Since images may have different color tables (palettes), you can save each image with its own color table. The source code for each image then includes its color table (see *File Formats* for a source code example). If you have used the same color palette for a series of images that are created for a given program, you'll want to use the next option.

##### **(2) One with color table**

This lets you create one copy of a color table (with the first image saved) so you don't re-save the same color table with each image.

##### **(3) No color tables**

If you've already saved images with a color table (as in 2, above) and are saving more images that use the same color table, or you simply don't need the color tables, select this option.

##### **(4) Assembly or C**

Select **Assembly** or **C** to choose the type of source code you want to generate. The source files are compatible with APW and are set to the proper language type. See *File Formats* for the file structures.

##### **(5) Mask**

To save a masked image, click the box beside **Mask**. **X** indicates it is selected. When selected, two copies of an image are saved as source code in a masked format. The first mask, the **And-Mask**, sets the bits of background-colored pixels to one. The second mask, the **Or-Mask**, resets the bits of background-colored pixels to zero.

Use the masks to display images on the screen without changing the screen area around them. If you need to display images over a background scene, or over each other, you will want to mask them. Otherwise, your image and its background color (the entire rectangular Edit Area) will copy over whatever is on the screen, cutting a rectangle out of the background scene.

When image masks are created, the Background Color is interpreted as a transparent color, so only the foreground colors are drawn.

If you want to use the Background Color as a foreground color in part of your image, you need to make special allowances since pixels within the image that are the Background Color are also interpreted as transparent. If you don't want them to be transparent, select a color for the background that isn't used in your image (of course, the background around your image must match the color you select). Since the Background Color becomes transparent, it doesn't matter which color you choose for

the background as long as the same color doesn't occur in the foreground (unless you do want part of your image to be transparent).

An alternative to changing the Background Color is to copy the background palette color over another unused palette color. Then use the newly created color as a foreground image color. The new color will not be transparent since its pixel value will be different than the background's pixel value (it's a different palette color and references a different color table entry).

The C programming example shown later copies an unmasked image to the screen. The assembly example displays a masked image. You can boot the *EXAMPLE.S16* file and see a series of the creature images overlapped slightly to demonstrate the effects of masking.

### **b. Save one options**

First, select **Save Clipboard One** and click the image you want to save in the Clipboard dialog box.

#### **(1) Assembly or Source**

As described above, select the type of source code you want to create.

#### **(2) Mask**

This is the same option described earlier. See section 5 above.

#### **(3) Color table**

Set this option if you want the image's color table converted to source and saved with it.

After clicking the name of the image, (or [Ok] if saving all), the disk save dialog box appears. Now, you can save the file like any other. If you need help with the save, refer to the *Disk Loads and Saves* section.

## **B. Generating Binary Image Files**

Every time you save Clipboard images in binary, you create a Binary file. This is the quickest and most disk-space-efficient way to save your Clipboard images.

You may also want to use this format to save images for programs, especially if you are working with large images.

Refer to the *Disk Loads and Saves* section if you need help with saving Clipboard images in binary. The *Binary File Format* section provides the information you need to use the images in your programs.

# C. File Formats

## 1. Source Code File Format

As mentioned earlier, it's the Clipboard images that you can convert to source code. The names you give your Clipboard images become their source code labels when they are converted to source code (examples are provided later).

Each source code image is saved with size information in the form of a rectangle. The image's rectangle is an array of 5 integers (rather than 4 like in the Tool Box reference manuals). The first 4 integers are identical to those required by some Quickdraw tool calls, so you can still pass a pointer to the rectangle for tool calls.

The Clipboard names are also used to make labels for rectangles, color tables, and masked images. The letters "Rect", "Ctbl", "Amsk", and "Omsk" are appended to names to make the additional labels when needed.

Each element of an image rectangle is described below:

Array element contents

- 0 top left Y coordinate - always 0 (zero)
- 1 top left X coordinate - always 0 (zero)
- 2 bottom right Y coordinate - image's pixel/byte height
- 3 bottom right X coordinate - image's pixel width
- 4 image's byte width

An image's rectangle defines its size. The rectangle's top left corner represents the top left corner of the image, and the bottom right corner represents the image's bottom right corner. Use this information to display an image directly to the screen (using the byte height from element 2, and the byte width from element 4). The programming examples given later demonstrate how. Image size rectangles are saved with all source code images since the size information is needed for *Graphics Supermarket* to load the image.

The color table is 32 bytes, or 16 integers, as defined in the Apple IIGS technical manuals. You can pass a color table pointer to Quickdraw tools as needed. This is demonstrated later in the programming examples.

Here are two sample listings of source code image files, one saved in C, and the other in assembly language. The C example was saved with the *mask* option off, and the assembly example with it on. The image is a make-believe creature drawn over a black background, however, the image shape is not important. It's simply used to

provide an example of both file structures. The creature image is also on the program disk so you can load and see it.

### Image saved as C source code

```
/* 241 */
/* 0 */

unsigned int CreatureRect[5] = (0,0,19,16,8);

char CreatureCTbl[32] = (
    0x00,0x00,0x30,0x08,0x41,0x08,0x2c,0x07,0x0f,0x00,
    0x29,0x00,0x00,0x0b,0x00,0x0d,0xa9,0x0f,0x53,0x09,
    0x13,0x05,0xdf,0x04,0xaf,0x0d,0x8f,0x07,0xcc,0x0c,
    0xff,0x0f);

char CreatureI[52] = (
    0x00,0x00,0x00,0x08,0x00,0x00,0x89,0x00,0x00,0x00,
    0x00,0x90,0x80,0x08,0x09,0x00,0x00,0x00,0x00,0x00,
    0x98,0x89,0x00,0x00,0x00,0x00,0x00,0x08,0x84,0x88,
    0x00,0x00,0x00,0x00,0x02,0x88,0x88,0x88,0x00,0x00,
    0x00,0x00,0x88,0x80,0x28,0x82,0x00,0x00,0x00,0x08,
    0x88,0x02,0x88,0x20,0x00,0x00,0x00,0x28,0x80,0x07,
    0x77,0x60,0x00,0x00,0x00,0x82,0x00,0x77,0x88,0x76,
    0x00,0x00,0x00,0x80,0x06,0x78,0x88,0x77,0x00,0x00,
    0x00,0x00,0x07,0x78,0x88,0x77,0x00,0x00,0x09,0x28,
    0x06,0x98,0x89,0x76,0x00,0x00,0x02,0x88,0x88,0x88,
    0x97,0x70,0x00,0x00,0x00,0x29,0x00,0x54,0x44,0x40,
    0x00,0x00,0x00,0x00,0x00,0x54,0x44,0x40,0x00,0x00,
    0x00,0x00,0x54,0x44,0x50,0x00,0x00,0x00,0x00,
    0x99,0x05,0x44,0x40,0x00,0x00,0x00,0x09,0xaa,0xa2,
    0x4a,0x2a,0x00,0x00,0x00,0x00,0x22,0xa2,0x22,0x29,
    0x00,0x00);
```

## Masked image saved as assembly source

```
;l1a1
;0
CreatureRect      ANOP
                  dc i2'0,0,19,16,8'
CreatureCTbl      ANOP
                  dc h'0000300841082c070f002900'
                  dc h'000b000da90f53091305df04'
                  dc h'afd08f07cc0cff0f'
CreatureAmsk      ANOP
                  dc h'ffffffff8ffff89fffffffff9f'
                  dc h'8ff8f9ffffffffffff9889ffff'
                  dc h'ffffffff08488fffffffff288'
                  dc h'8888ffffffff888f2882ffff'
                  dc h'fff888f2882fffffffff288f17'
                  dc h'776ffffffff62ff78076ffff'
                  dc h'f8f6788877ffffffff778'
                  dc h'8877fffff928f6988976ffff'
                  dc h'f28888889777ffffffff29f54'
                  dc h'444ffffffffffff54444ffff'
                  dc h'fffff54445fffffffff99f5'
                  dc h'444fffffffff9aaa24a2affff'
                  dc h'fffff22a22229ffff'
CreatureOmsk      ANOP
                  dc h'000000080000890000000090'
                  dc h'80889000000000098890000'
                  dc h'0000008848800000000288'
                  dc h'88880000000888028820000'
                  dc h'00088802882000000288007'
                  dc h'77600000082007788760000'
                  dc h'00800678887700000000778'
                  dc h'887700000928069889760000'
                  dc h'02888888977000000290054'
                  dc h'444000000000544400000'
                  dc h'0000054445000000009905'
                  dc h'44400000009aaa24a2a0000'
                  dc h'000022a222290000'
```

You can use the C statement, `SetColorTable(0,CreatureCTbl);`, to pass a pointer to the color table of the image in the listing to a Quickdraw tool. The tool sets color table zero to the image's color table. The *Apple IIGS Tool Box Reference Manuals* contain detailed tool call descriptions.

Each sample listing has comments, called *descriptors* containing numeric values (`/*241*/` and `;l1a1` for example). Each source file has a descriptor, and each image has its own descriptor. The descriptors are boolean words (16 bit) defined as follows:

Bit #'s	Description
FileDesc:	0-3 number of images in file (1-15)
	4 reserved
	5-6 color tables, 00=none; 10=one; 11=each image has table
	7 mask; 1=image's masked
	8-9 10 = assembly source; 01 = C source
	10-15 reserved
ImageDesc:	0 1 = dithered if 640 mode
	1 reserved (always 1; saved in rows as bit mapped rect)
	2-6 reserved
	7 1 = 640 mode; 0 = 320 mode
	8-11 background color (0-15)
	12-15 reserved

The descriptors are used by *Graphics Supermarket* to reload source files, so do not change source files if you want to reload and edit them.

A good general strategy is to save all images in binary for editing (it's more space efficient too) and generate source only of final images for programs. If you need to edit an image later, you can load and edit the binary version, re-save it, and generate new source if needed. You'll always have your binary files available for editing images and you can then edit the source files. You can change them, delete unwanted data, append them to other source files, combine them into shape files, or use them any way you wish for programming.

## 2. Binary File Format

### a. Picture Files

You can create binary full screen image files with the **Save Picture** or **Save Picture Binary** menu items. The file formats are identical. The only difference is the Prodos file type. The **Save Picture** file type is \$C1 and the **Save Picture Binary** file type is 6.

Picture files consist of 32768 bytes structured as follows:

32000	picture image
256	SCBs (Scan line Control Bytes), the 1st 200 used, last 56 reserved
512	16 color tables of 32 bytes each (tables 0-15)

Refer to the GS technical manuals for detailed information on SCBs and color tables.

*Graphics Supermarket* saves the Color Palette for the picture as color table zero. You can find the start address of the color table by adding the picture length (32000) plus the SCBs' length (256) to the program's start address. For example, if you load the picture file into memory beginning at address 8000, the address where the color table begins is 40256 (8000 + 32000 + 256).

Three bytes of color table 15 are written over with the background color, mode, and a flag that indicates the file is a *Graphics Supermarket* picture. The 3 bytes are located as follows:

Picture start address + \$7FFA = Background Color (0-15)

+ \$7FFC = Mode; 0 = 320, \$80 = 640, \$81 = dither 640

+ \$7FFE = Flag; 63

### b. Clipboard Files

Binary Clipboard image files contain up to 15 images. A program can load binary image files and use the images as desired. Following is a description of the Clipboard binary file structure.

#### Binary Clipboard file structure

	Number of Bytes Used	Description
	4	file header
Image Record {	56	image header
	1-32000	image
	6	reserved

Each binary image file has one 4 byte *file header*. The number of *image records* in the file equals the number of images as each image has an image record. The image's record size depends on the image's size. The *image header* size is always 56 bytes, and there are always 6 reserved bytes at the end of each image. So, the image record size will vary according to the number of bytes the image itself uses.

The first 2 bytes of the file header flag verify the file as a Clipboard image file. They should always equal \$0099. The next 2 bytes are the file descriptor bytes. Bits 0-3 of the first byte contain the number of images in the file (0-15). The remaining bits (of both bytes) are reserved.

Following is a description of the image header:

Byte Offset	Use
0-1	integer equalling image record size minus 2
2	reserved; always 1 for saved as bit mapped rect-rows
3	mode byte; 0=320, \$80=4 color 640, \$81=dithered 640
4	name/label length byte; for printing name string
5-20	name/label; space for 15 ASCII chars +NULL (zero)
21-52	color table (32 bytes stored as 16 integers)
53	image byte width (1-160)
54	image byte height (1-200)
55	background color (0-3 in 4 color 640 mode else 0-15)

The first 2 bytes of the image header contain the total byte size of the image's record. The size is an integer that you can use as an offset (or index) to the next image in the file. It doesn't take into account the 2 bytes it occupies, so be sure to add 2 when computing the address of the next image. For example, if the address of the current image record is 8000 and its record size is 200, the address of the next image's record is 8202 (8000 + 200 + 2), and the address of the image itself is 8258 (8202 + 56, the image header size).

Don't forget, an integer is stored in memory with the most significant byte at the highest address. How you reference it depends on the language you use. For example, assuming that the current image's record address is still the 8000 used above, you can get the integer length from BASIC with the statements

```
100 RSADR = 8000
```

```
110 RS = PEEK(RSADR + 1) * 256 + PEEK(RSADR) + 2
```

```
120 NEXT = RSADR + RS 'next image's record address
```

where "RSADR" is the address of the current image's record size, "RS" is the record size, and "NEXT" is the address of the next image's record.

Use an image's width and height to display the image directly to screen memory (\$E12000-\$E17CFF). Reset (set to 0) bit 0 of the mode byte if you use it to setup the display mode. Bit zero is used as a flag by *Graphics Supermarket* to indicate which 640 mode format (16 dithered colors or 4 solid colors) was used to draw an image.

The Background Color also reflects the background pixel pattern of an image (this could change in future versions that use more complex background patterns). It's range is 0 to 15 in the 320 mode and dithered 640 mode, and 0 to 3 in the 4 color 640 mode. If the background color byte is 2 (bit pattern = 0000 0010), the background bit pattern for an entire byte looks as follows:



320 and dithered 640 mode:

0010 0010 two 4 bit pixels (simulated in dithered mode)

4 color 640 mode:

10 10 10 10 four 2 bit pixels

*Note: As described earlier, you can save images as source code with a mask for displaying them to the screen. You can't do this with a binary save. However, you could save image's as source code (with the masks if desired) and compile them into your own binary file.*

### 3. Programming Examples

This section contains programming examples for using images created with *Graphics Supermarket*. Examples are provided for C, assembly language, and Applesoft BASIC. All of the examples are also on your program disk. You can compile and experiment with the C and assembly language examples, or RUN the Applesoft BASIC examples to try them out.

#### a. C examples

The following program fragment starts up 5 IIGS tools and draws the sample creature image created by *Graphics Supermarket* and saved as C source code. The source code was shown earlier in this section and is included in the example program fragment. This listing is the *EXAMPLE.C* file on your program disk.

The input parameters to the draw routine, *DrawImage*, are the image's width and height in bytes, the X and Y screen coordinates where you want the image displayed, and a pointer to the image's bit map (the image itself).

#### C example

```
#include <locator.h>
#include <memory.h>
#include <misctool.h>
#include <quickdraw.h>
#include <event.h>

#define MODE 0           /* 320 mode */
#define XMAXCURS 320    /* 320 for 320 mode, 640 for 640 mode */
#define PIXBYT 2        /* 4 for 640 mode */
char *ZPPtr, **ZPHand; /* direct pg ptr, hndl */
unsigned int MasterID; /* this prog's master ID */
unsigned int AppID;    /* ID for requesting additional mem */

struct event {          /* Event Mgr-TaskMaster rec */
    int what;
    char *message;
    int whenH;
    int whenL;
    int whereY;
    int whereX;
    int modifiers;
    long taskData;
    long taskMask;
} EventRec;
```

```
unsigned int CreatureRect[5] = (0,0,19,16,8);
```

```
char CreatureCtbl[32] = (  
    0x00,0x00,0x30,0x08,0x41,0x06,0x2c,0x07,0x0f,0x00,  
    0x29,0x00,0x00,0x0b,0x00,0x0d,0xa9,0x0f,0x53,0x09,  
    0x13,0x05,0xdf,0x04,0xaf,0x0d,0x8f,0x07,0xcc,0x0c,  
    0xff,0x0f);
```

```
char Creature[152] = (  
    0x00,0x00,0x00,0x08,0x00,0x00,0x89,0x00,0x00,0x00,  
    0x00,0x90,0x80,0x08,0x09,0x00,0x00,0x00,0x00,0x00,  
    0x98,0x89,0x00,0x00,0x00,0x00,0x00,0x08,0x84,0x88,  
    0x00,0x00,0x00,0x00,0x02,0x88,0x88,0x88,0x00,0x00,  
    0x00,0x00,0x88,0x80,0x26,0x82,0x00,0x00,0x00,0x08,  
    0x88,0x02,0x88,0x20,0x00,0x00,0x00,0x26,0x80,0x07,  
    0x77,0x60,0x00,0x00,0x00,0x82,0x00,0x77,0x88,0x76,  
    0x00,0x00,0x00,0x80,0x06,0x78,0x88,0x77,0x00,0x00,  
    0x00,0x00,0x07,0x76,0x88,0x77,0x00,0x00,0x09,0x28,  
    0x06,0x98,0x89,0x76,0x00,0x00,0x02,0x88,0x88,0x88,  
    0x97,0x70,0x00,0x00,0x00,0x29,0x00,0x54,0x44,0x40,  
    0x00,0x00,0x00,0x00,0x00,0x54,0x44,0x40,0x00,0x00,  
    0x00,0x00,0x00,0x54,0x44,0x50,0x00,0x00,0x00,0x00,  
    0x99,0x05,0x44,0x40,0x00,0x00,0x00,0x09,0xaa,0xa2,  
    0x4a,0x2a,0x00,0x00,0x00,0x00,0x22,0xa2,0x22,0x29,  
    0x00,0x00);
```

```
char CreatureMask[152] = (  
    0xff,0xff,0xff,0xf8,0xff,0xff,0x89,0xff,0xff,0xff,  
    0xff,0x9f,0x8f,0xf8,0xf9,0xff,0xff,0xff,0xff,0xff,  
    0x98,0x89,0xff,0xff,0xff,0xff,0xf8,0x84,0x88,  
    0xff,0xff,0xff,0xff,0xf2,0x88,0x88,0x88,0xff,0xff,  
    0xff,0xff,0x88,0x8f,0x26,0x82,0xff,0xff,0xff,0xf8,  
    0x88,0xf2,0x88,0x2f,0xff,0xff,0xff,0x26,0x8f,0xf7,  
    0x77,0x6f,0xff,0xff,0xff,0x82,0xff,0x77,0x88,0x76,  
    0xff,0xff,0xff,0x8f,0xf6,0x78,0x88,0x77,0xff,0xff,  
    0xff,0xff,0xf7,0x78,0x88,0x77,0xff,0xff,0xf9,0x28,  
    0xf6,0x98,0x89,0x76,0xff,0xff,0xf2,0x88,0x88,0x88,  
    0x97,0x7f,0xff,0xff,0xff,0x29,0xff,0x54,0x44,0x4f,  
    0xff,0xff,0xff,0xff,0xff,0x54,0x44,0x4f,0xff,0xff,  
    0xff,0xff,0xff,0x54,0x44,0x5f,0xff,0xff,0xff,0xff,  
    0x99,0xf5,0x44,0x4f,0xff,0xff,0xff,0xf9,0xaa,0xa2,  
    0x4a,0x2a,0xff,0xff,0xff,0xff,0x22,0xa2,0x22,0x29,  
    0xff,0xff);
```

```
/*  
/* DRAW IMAGE  
*/
```

```
void  
DrawImage(Width, Height, XLoc, YLoc, ImgPtr)  
    unsigned int Width, Height, XLoc, YLoc; /* img's wdt, hgt, screen X,Y */  
    char *ImgPtr; /* pointer to image */  
(  
    unsigned int i, w, h; /* temp index, wdt, hgt vars */  
    char *ScrPtr; /* to point to screen */  
    /* get pointer to screen */  
    ScrPtr = ((char*)(0xE12000)) + (YLoc * 160) + (XLoc / PIXBYT);
```

```

i = 0; /* image byte index */
for(h=0; h < Height; h++) { /* image bytes to screen loop */
    for(w=0; w < Width; w++, i++) /* do an image row loop */
        *(ScrPtr + w) = *(ImgPtr + i); /* image byte to screen */
    ScrPtr += 160; /* down a screen line */
}
)

main()
(
/*
/*          START UP TOOLS
*/

    TLStartUp(); /* Tool Loc */
    MasterID = MMStartUp(); /* Mem Mgr */
    AppID = MasterID | 0x0100; /* create ID for appl use */
    MTStartUp(); /* Misc Tools */
    ZPHand = NewHandle(0x400L, AppID, 0xC005, 0L); /* get DP mem */
    ZPPtr = *ZPHand; /* deref zero pg ptr */
    QDStartUp((int)ZPPtr, MODE, 160, AppID); /* Quickdraw */
    EMStartUp((int)ZPPtr+0x300, 20, 0, XMAXCURS, 0, 200, AppID); /* Event Mgr */

/*
/*          SHOW THE IMAGE
*/

    SetColorTable(0, CreatureCTbl); /* use images colors */

/* draw image to screen */
    DrawImage(CreatureRect[4], CreatureRect[2], 152, 85, Creature);

    while(!GetNextEvent(2, &EventRec)); /* wait for button press */

/*
/*          SHUT DOWN TOOLS - QUIT
*/

    GrafOff(); /* hires off */
    EMShutDown(); /* shut'em down */
    QDShutDown();
    MTShutDown();
    DisposeAll(AppID); /* release ZP memory */
    MMShutDown(MasterID);
    TLShutDown();
)

```

## b. Assembly examples

This program fragment does the assembly language equivalent of the previous C program fragment, except the creature image is masked to the screen. After a single image is drawn near the center of the screen, a series of images are drawn slightly overlapped to demonstrate how masking draws only an image's foreground colors. If they were not masked, each successive image that is drawn would chop out a rectangular portion of the previous image.

The listing is the *EXAMPLE.ASM* file on your program disk. The input parameters to the draw routine, *DrawImage*, are the image's width and height in bytes, the X and Y

screen coordinates where you want the image displayed, and pointers to the image's masks. The parameters are pushed on the stack and passed to the draw routine.

Following the program fragment is a listing of the macro file used for tool calls (it's copied into the program with the *mcopy example.macros* line near the start). The macro file, *EXAMPLE.MACROS*, and the compiled version of this example, *EXAMPLE.S16*, is included on your program diskette. If you want to, you can boot *EXAMPLE.S16* and see the end result of this example. You can execute it from the program launcher or APW.

### Assembly language example

```

        absaddr on
        KEEP example
        mcopy example.macros      has tool macro calls, etc

GSEExample    START

ScreenMode    equ 0                320 mode, $80 = 640 mode
MaxWdt        equ 320              else 640 for the 640 mode

        phk                        data bank=code bank for abs adressng
        plb

;-----
; begin tool startup

        _TLStartup                Tool Loc
        PushWord #0                Memory Hgr
        _MMStartup
        pla
        sta ProgID                save ID
        _MTStartup                Misc tools

;-----
; get memory for direct pages

        PushLong #0                space for handle
        PushLong ##400             4 pages
        PushWord ProgID
        PushWord ##C001            locked, fixed, fixed bank
        PushLong #0
        _NewHandle                 get DP memory
        pla                        save handle to it
        sta 0
        sta DPHandle
        pla
        sta 2
        sta DPHandle+2
        lda [0]                    get direct page
        sta 4                      and save

;-----
; continue tool start up

        pha
        PushWord #ScreenMode

```

I

```
PushWord #160          number of bytes per line
PushWord ProgID
_QDStartup            Quick Draw
lda 4                  DP to use = QuickDraw DP +$300
clc
adc #$300
pha
PushWord #20           queue size
PushWord #0           X min
PushWord #MaxWdt      X max
PushWord #0           Y min
PushWord #200         Y max
PushWord ProgID
_EMStartup            Event Mgr
```

-----

; show the Creature image

```
ldx #152              set up creature's X scrn loc
stx CreatureX
ldx #85                and Y loc
stx CreatureY
jsr DrawCreature      set up params, call draw rout

ldx #0                 overlap a few to show masking effects
stx CreatureX
ldx #140
stx CreatureY
lda #39                do 39
```

OverLap

```
pha
jsr DrawCreature
lda CreatureX
clc
adc #8
sta CreatureX
pla
dec a
bne OverLap
```

-----

; wait for mouse down event

```
WaitDown              PushWord #0          wait for down event
                     PushWord #10
                     PushLong #EventRec

                     _GetNextEvent
                     pla
                     beq WaitDown
```

-----

;Shut down tools - Quit

```
_EMShutDown
_QDShutDown
_MTShutdown
```

```

PushLong DPHandle
_DisposeHandle
PushWord ProgID
_MMShutdown
_TLShutdown
_Quit QuitParams

```

```

;-----
; Draw the example creature image, similar setups can be
; used for other images (or set up a loop and do an array of images)

```

```

CreatureX      ds 2
CreatureY      ds 2
DrawCreature   PushWord #0           use color table 0
               PushLong #CreatureCtbl image's color table
               _SetColorTable      set it up
               PushWord CreatureX   X screen loc
               PushWord CreatureY   Y screen loc
               PushWord CreatureRect+8 Width
               PushWord CreatureRect+4 Height
               PushWord #CreatureDmask pointer to image or-mask (2 byte addr)
               PushWord #CreatureAmask pointer to image and-mask
               jmp DrawImage        show the image (leaves rts on stack)...
;                                     could leave out the JMP instruction
;                                     in this case and let it fall
;                                     through to draw subrout. since
;                                     it's just below...but that wouldn't
;                                     do for calls from other locations
;                                     so it's left in for example purposes.

```

```

;-----
; Routine to draw an image saved in rows...
; Push the following on the stack as listed from top to bot
;   screen x loc (word)
;   screen y loc (word)
;   image width (word)
;   image height (word)
;   image pointer(long address)
; Note: image width must be a multiple of 2 (2, 4, 6 ...); data bank=prog bank

```

```

AndMask        equ 1           image And-Mask Ptr; DP offset
OrMask         equ AndMask+2   image Or-Mask Ptr
Height         equ OrMask+2    image byte height
Width          equ Height+2    image byte width
YLoc           equ Width+2     screen Y location
XLoc           equ YLoc+2      screen X location
DrawImage      tsc             get current stack
               phd             save direct page
               tcd             and use parms as new direct page
               lda XLoc        get X and calc screen byte location
               lsr a           divide by 2
;               lsr a           need this if 640 mode
               pha            save horizontal byte offset
               ldy YLoc        get Y and ver offs
               ldx #160        Number of bytes per line
               jsr MultXY
               cfc

```

```

I      adc     1,s      add hor and ver offs
      plx      clean stack
      tax      X register has offset
      lsr      Width   divide width by 2, stores dbl bytes
      pei     (Width)  save the width
      ldy     #0       image index = 0 = 1st img byte

ToScreenLp
tsl2   phx      save offset
      lda     $E12000,x get screen byte
      and    (AndMask),y reset img foregrd bits to zero
      ora    (OrMask),y  set img foregrd bits to one
      sta     $E12000,x  back to the screen
      inx      move over 2 columns
      inx
      iny      skip 2 image bytes
      iny
      dec     Width    width counter
      bne    tsl2
      pla      get orig offset
      clc
      adc     #160     down a scrn line
      tax      as new index
      cmp     #32000   bottom of screen?
      bcs    Exit     if so, stop here
      lda     1,s
      sta     Width    reset width ctr
      dec     Height   all rows done?
      bne    ToScreenLp no, continue

Exit   pla      clean stack
      pid     get orig direct page
      tsc     skip over params on stack
      clc
      adc     #12      total param bytes
      tcs
      rts      done

-----
; Multiply two 16 bit numbers passed in the X,Y regs
; Result returned in accumulator

MultXY   lda     #0
      sty     mop2
      stx     mop1
m1       ldx     mop1
      beq     m3
      lsr     mop1
      bcc     m2
      clc
      adc     mop2
m2       asl     mop2
      bra     m1
m3       rts
mop1     ds 2
mop2     ds 2

QuitParams dc i4'0'
          dc i'00000'      not restartable

```

```

P  EventRec      anop                record for GetNextEvent tool
   EventWhat     ds 2
   EventMsg      ds 4
   EventWhen     ds 4
   EventWhere    ds 4
   EventMods     ds 2

DPHandle        ds 4                direct page handle
ProgID          ds 2                this program's ID

CreatureRect    ANOP                image rectangle
               dc i'0,0,19,16,8'

CreatureCTbl    ANOP                image color table
               dc h'0000300841082c070f002900'
               dc h'000b00da90f53091305df04'
               dc h'af0d8f07cc0cff0f'

CreatureAmsk    ANOP                image And-Mask
               dc h'ffffffff8ffff89ffffffff9f'
               dc h'8ff8f9fffffffff9889ffff'
               dc h'ffffffff88400ffffffff288'
               dc h'8888ffffffff888f2882ffff'
               dc h'fff888f2882ffffffff288ff7'
               dc h'776ffffffff82ff778876ffff'
               dc h'ff8ff6788877ffffffff778'
               dc h'8877ffffffff928f6988976ffff'
               dc h'f2888888977ffffffff29ff54'
               dc h'444ffffffffffff54444ffff'
               dc h'ffffffff54445fffffffff99f5'
               dc h'444fffffffff9aa24a2affff'
               dc h'ffff22a22229ffff'

CreatureOmsk    ANOP                image Or-Mask
               dc h'000000080000890000000090'
               dc h'800809000000000098890000'
               dc h'0000008848800000000288'
               dc h'88880000000888028820000'
               dc h'00088802882000000288007'
               dc h'77600000082007788760000'
               dc h'0800678887700000000778'
               dc h'8870000928069889760000'
               dc h'0288888977000000290054'
               dc h'44400000000005444400000'
               dc h'000005444500000009905'
               dc h'4440000009aa24a2a0000'
               dc h'000022a222290000'

```

### Macro file for the above example

```

MACRO
&lab _EMStartUp
&lab idx #0206
jsl $E10000
MEND

```

```

MACRO
&lab _EMShutDown
&lab idx #0306
jsl $E10000
MEND

```

```

MACRO
&lab _GetNextEvent
&lab idx #0A06
jsl $E10000
MEND

```



```

# MACRO
&lab _TLStartUp
&lab ldx #0201
jsl $E10000
MEND

MACRO
&lab _TLShutDown
&lab ldx #0301
jsl $E10000
MEND

MACRO
&lab _MMStartUp
&lab ldx #0202
jsl $E10000
MEND

MACRO
&lab _MMShutDown
&lab ldx #0302
jsl $E10000
MEND

MACRO
&lab _NewHandle
&lab ldx #0902
jsl $E10000
MEND

MACRO
&lab _DisposeHandle
&lab ldx #0102
jsl $E10000
MEND

MACRO
&lab _MTStartUp
&lab ldx #0203
jsl $E10000
MEND

MACRO
&lab _MTShutDown
&lab ldx #0303
jsl $E10000
MEND

MACRO
&lab _QUIT &params
&lab jsl $E100A8
dc i2"$29"
dc i4"&params"
MEND

MACRO
&lab _QDStartUp
&lab ldx #0204
jsl $E10000
MEND

MACRO
&lab _QDShutDown
&lab ldx #0304
jsl $E10000
MEND

MACRO
&lab _GrafOn
&lab ldx #4+256*10
jsl $E10000
MEND

MACRO
&lab _GrafOff
&lab ldx #0804
jsl $E10000
MEND

MACRO
&lab _InitCursor
&lab ldx #0CA04
jsl $E10000
MEND

MACRO
&lab _SetColorTable
&lab ldx #0E04
jsl $E10000
MEND

MACRO
&lab pushlong &addr,&offset
&lab ANOP
LCLC &C
LCLC &REST
&C AMID &addr,1,1
MEND

AIF "&C"="#",.immediate
AIF "&C"="I",.zeropage
AIF C:&offset=0,.nooffset
AIF "&offset"="s",.stack
pushword &addr+2,&offset
pushword &addr,&offset
NEXIT

.nooffset
pushword &addr+2
pushword &addr
NEXIT

.immediate
&REST AMID &addr,2,L:&addr-1
dc I1'$F4',I2'(&REST)1-16'
dc I1'$F4',I2'&REST'
NEXIT

.stack
pushword &addr+2,s
pushword &addr+2,s
NEXIT

.zeropage
ldy #&offset+2
pushword &addr,y
ldy #&offset
pushword &addr,y
MEND

MACRO
&lab pushword &sysopr
&lab ANOP
AIF c:&sysopr=0,.b
LCLC &C
LCLC &REST
LCLA &BL
&C AMID "&sysopr",1,1
&BL ASEARCH "&sysopr"," ",1
AIF &BL>0,.a
&BL SETA L:&sysopr+1
.a
&REST AMID "&sysopr",2,&BL-2
AIF ("&C"="#").AND.(S:LONGA),.immediate
lda &sysopr
pha
NEXIT

.b
pha
NEXIT

.immediate
dc I1'$F4',I2'&REST'
MEND

```

### c. Basic Examples

This section contains examples that show how to display both picture and Clipboard images from Applesoft BASIC. Examples are given for using Applesoft under Prodos and Dos 3.3.

All the examples work with binary image files, so use the **Save Picture Binary** menu item when saving pictures, and save Clipboard images with the binary option selected.

The following example displays a full screen picture from Applesoft running under Prodos. It's a listing of the *PDOS.SHOWPIC* file included on your *Graphics Supermarket* program disk.

### Prodos show picture

```
99  REM SETUP MOVE->AUX CALL
100 POKE 768,56: POKE 769,76
110 POKE 770,17: POKE 771,195
115 D$ = CHR$(4)
116 ONERR GOTO 10000
120 GOSUB 1200: REM SHADOW ON
130 HOME : PRINT "      <GRAPHICS SUPERMARKET EXAMPLE>": PRINT
135 PRINT "DISPLAYS A STANDARD IIGS PICTURE...": PRINT
140 INPUT "PICTURE'S PATHNAME: ";F$
240 GOSUB 1300: REM ERASE
250 GOSUB 1210: REM HIRES ON
260 GOSUB 1400: REM LOAD PIC
270 GET A$: PRINT CHR$(13)
290 GOSUB 1220: REM HIRES OFF
300 GOTO 130
997 REM
998 REM COPY MAIN->AUX
999 REM SA=START,EA=END,DA=DEST
1000 POKE 60,SA - ( INT (SA / 256) * 256)
1010 POKE 61, INT (SA / 256)
1020 POKE 62,EA - ( INT (EA / 256) * 256)
1030 POKE 63, INT (EA / 256)
1040 POKE 64,DA - ( INT (DA / 256) * 256)
1050 POKE 67, INT (DA / 256)
1060 CALL 768
1070 RETURN
1198 REM
1199 REM SHADOW ON
1200 POKE 49205,0: RETURN
1208 REM
1209 REM SUPER HIRES ON
1210 POKE 49193,163: RETURN
1218 REM
1219 REM SUPER HIRES OFF
1220 POKE 49193,1: RETURN
1298 REM
1299 REM ERASE SUPER HIRES->BLACK
1300 POKE 230,32: CALL 62450
1310 POKE 230,64: CALL 62450
1320 SA = 8192:EA = SA + 16383:DA = SA: GOSUB 1000
1330 DA = SA + 16384: GOSUB 1000
1335 RETURN
1399 REM LOAD PICTURE
1400 PRINT D$;"BLOAD";F$;"A8192,L28000"
1410 SA = 8192:EA = 36192:DA = 8192
```

```

1420 GOSUB 1000
1430 PRINT D$;"BLOAD";F$;" ,A8192,B28000,L14288"
1440 EA = SA + 4288;DA = SA + 28000
1450 GOSUB 1000: RETURN
9999 REM ERROR TRAP
10000 GOSUB 1220: REM HIRES OFF
10010 PRINT"ERROR ";PEEK(222);" IN LINE ";PEEK(218)+PEEK(219)*256
10020 END

```

The next listing is the *PDOS.SPLITPIC* file included on your program disk. You can use it to split a picture in half for displaying in Applesoft BASIC running under Dos 3.3. The example following this one displays the two halves.

To prepare a picture for use with Dos 3.3, RUN the *PDOS.SPLITPIC* file under Prodos Applesoft BASIC. Enter your picture's pathname. The picture is loaded and re-saved as two new files (leaving the original file on the disk). The character "1" is automatically appended to the pathname of the file that contains the top half of the picture, and a "2" is appended to the file's pathname that contains the bottom half.

Now, use the *CONVERT* utility on your Prodos8 system disk to copy the two picture files, and the *DOS3.3SHOWPIC* example file listed next, to a Dos 3.3 disk. Then RUN *DOS3.3SHOWPIC* to load the two halves and display the picture.

### Split picture for Dos 3.3.

```

100 D$ = CHR$(4)
130 HOME : PRINT " (GRAPHICS SUPERMARKET EXAMPLE)"; PRINT
135 PRINT "SPLITS A PICTURE IN HALF FOR THE DOS 3.3SHOWPIC EXAMPLE..."; PRINT
140 INPUT "PICTURE'S PATHNAME: ";F$
190 PRINT : PRINT "LOADING: ";F$
200 PRINT D$;"BLOAD";F$ ,A8192,L18000
205 PRINT " SAVING: ";F$;"1"
210 PRINT D$;"BSAVE";F$;"1,A8192,L18000"
215 PRINT " SAVING: ";F$;"2": PRINT
220 PRINT D$;"BLOAD";F$;" ,A8192,B18000,L14288"
230 PRINT D$;"BSAVE";F$;"2,A8192,L14288"
240 PRINT "TWO FILES SAVED!"
250 END

```

The *DOS3.3SHOWPIC* file on your *Graphics Supermarket* program disk is listed below. You can use the previous example to prepare a picture for displaying, then display it from Applesoft BASIC with this example.

### Dos 3.3 show picture

```

99 REM SETUP MOVE->AUX CALL
100 POKE 768,56: POKE 769,76
110 POKE 770,17: POKE 771,195
115 D$ = CHR$(4)

```

```

116 ONERR GOTO 10000
120 GOSUB 1200: REM SHADOW ON
130 HOME : PRINT " (GRAPHICS SUPERMARKET EXAMPLE)": PRINT
140 PRINT "DISPLAYS A STANDARD IIGS PICTURE..."
150 PRINT : PRINT "ENTER THE PICTURE'S NAME ONLY. A '1' AND '2' DIGIT IS AUTOMATICALLY
APPENDED TO THE NAME WHEN LOADING EACH HALF OF THE PICTURE.": PRINT
160 INPUT "PICTURE'S PATHNAME: ";F$
240 GOSUB 1300: REM ERASE
250 GOSUB 1210: REM HIRES ON
260 GOSUB 1350: REM LOAD PIC
270 GET A$
290 GOSUB 1220: REM HIRES OFF
300 GOTO 130
997 REM
998 REM COPY MAIN->AUX
999 REM ERROR TRAP
1000 POKE 60,SA - ( INT (SA / 256) * 256)
1010 POKE 61, INT (SA / 256)
1020 POKE 62,EA - ( INT (EA / 256) * 256)
1030 POKE 63, INT (EA / 256)
1040 POKE 64,DA - ( INT (DA / 256) * 256)
1050 POKE 67, INT (DA / 256)
1060 CALL 768
1070 RETURN
1198 REM
1199 REM SHADOW ON
1200 POKE 49205,0: RETURN
1208 REM
1209 REM SUPER HIRES ON
1210 POKE 49193,163: RETURN
1218 REM
1219 REM SUPER HIRES OFF
1220 POKE 49193,1: RETURN
1298 REM
1299 REM ERASE SUPER HIRES->BLACK
1300 POKE 230,32: CALL 62450
1310 POKE 230,64: CALL 62450
1320 SA = 8192:EA = SA + 16383:DA = SA: GOSUB 1000
1330 DA = SA + 16384: GOSUB 1000
1335 RETURN
1349 REM LOAD PICTURE
1350 PRINT D$;"BLOAD"F$*1,A8192
1360 SA = 8192:EA = SA + 18000:DA = 8192
1370 GOSUB 1000
1400 PRINT D$;"BLOAD"F$*2,A8192
1410 SA = 8192:EA = SA + 14288:DA = SA + 18000
1420 GOSUB 1000
1430 RETURN
10000 GOSUB 1220
10010 PRINT"ERROR ";PEEK(222);" IN LINE ";PEEK(218)+PEEK(219)*256
10020 END

```

The next example is a listing of the *PDOS.SHOWCLIP* file. You can RUN it from Applesoft BASIC to display Clipboard image's saved in binary. There is also a Dos 3.3 version of this file, *DOS3.3SHOWCLIP*, included on your program disk. You can

convert it and Clipboard binary image files to Dos 3.3 disks and display images from Applesoft running under Dos 3.3. The only difference in the two files is the load subroutine beginning at line 1400.

You do have Clipboard size and memory limits to observe for these examples to work correctly. There's enough memory to load a binary file that's about 25K from Applesoft. You can use the **Clipboard Byte Size** item under the **STATUS** menu to determine what size your Clipboard image file will be. If you try to load a Clipboard file that's too large, you might crash the system (if this happens, you should re-boot the computer to reinitialize it).

You can safely load all the Clipboard image files contained on your *Graphics Supermarket* program diskette (some are *Misc*, *Computer*, *Sports*, and *Weather*).

### Prodos show Clipboard

```
99  REM SETUP MOVE->AUX CALL
100 POKE 768,56: POKE 769,76
110 POKE 770,17: POKE 771,195
115 DIM IMADR(15),NAME$(15)
116 ONERR GOTO 10000
120 GOSUB 1200: REM SHADOW ON
130 HOME : PRINT " (GRAPHICS SUPERMARKET EXAMPLE)": PRINT
135 PRINT "DISPLAYS CLIPBOARD IMAGES...": PRINT
140 HIMEM: 36352: REM KEEPS PRODOS BFRS HI
150 INPUT "IMAGE'S PATHNAME: ";F$
230 D$ = CHR$(4)
240 GOSUB 1300: REM ERASE
250 GOSUB 1400: REM LOAD IMAGE FILE
260 GOSUB 1500: REM GET NAMES
270 HOME : PRINT : FOR X = 1 TO TTL
275 S$ = " " : IF X < 10 THEN S$ = " "
280 PRINT S$;X;") "NAME$(X)
290 NEXT X: PRINT S$;TTL + 1;") LOAD ANOTHER FILE": PRINT
300 PRINT "SHOW WHICH IMAGE(1-";TTL + 1");":
310 INPUT N: IF N < 1 OR N > TTL THEN 130
315 PRINT N
330 GOSUB 1600: REM SHOW IMAGE
340 GET A$: PRINT CHR$(13)
345 GOSUB 1220: REM HIRES OFF
350 GOTO 270
997 REM
998 REM COPY MAIN->AUX
999 REM SA=START,EA=END,DA=DEST
1000 POKE 60,SA - ( INT (SA / 256) * 256)
1010 POKE 61, INT (SA / 256)
1020 POKE 62,EA - ( INT (EA / 256) * 256)
1030 POKE 63, INT (EA / 256)
1040 POKE 64,DA - ( INT (DA / 256) * 256)
1050 POKE 65, INT (DA / 256)
1060 CALL 768
1070 RETURN
1198 REM
```

```

1199 REM SHADOW ON
1200 POKE 49205,0: RETURN
1208 REM
1209 REM SUPER HIRES ON
1210 POKE 49193,163: RETURN
1218 REM
1219 REM SUPER HIRES OFF
1220 POKE 49193,1: RETURN
1298 REM
1299 REM ERASE SUPER HIRES->BLACK
1300 POKE 230,32: CALL 62450
1310 POKE 230,64: CALL 62450
1320 SA = 8192:EA = SA + 16383:DA = SA: GOSUB 1000
1330 DA = SA + 16384: GOSUB 1000
1335 RETURN
1400 PRINT D$;"BLOAD";F$;"",A8192,L28000"
1410 RETURN
1499 REM GET TOT IMAGES & NAMES
1500 TTL = PEEK (8194)
1510 A = 8196: REM 1ST IMAGE REC
1520 FOR X = 1 TO TTL
1530 IMADR(X) = A:NAME$(X) = ""
1540 FOR Y = A + 5 TO A + 19
1550 NAME$(X) = NAME$(X) + CHR$( PEEK (Y))
1560 NEXT Y
1570 A = A + ( PEEK (A) + ( PEEK (A + 1) * 256) + 2)
1580 NEXT X
1590 RETURN
1599 REM SHOW IMAGE
1600 W = PEEK (IMADR(N) + 53)
1610 H = PEEK (IMADR(N) + 54)
1620 HOME
1630 PRINT "X BYTE LOC(0-";160 - W;");": INPUT X
1640 IF X + W > 159 THEN X = 159 - W
1650 IF X < 0 THEN X = 0
1660 PRINT "Y BYTE LOC(0-";200 - H;");": INPUT Y
1670 IF Y + H > 199 THEN Y = 199 - H
1680 IF Y < 0 THEN Y = 0
1685 GOSUB 1210: REM HIRES ON
1690 DA = Y * 160 + X + 8192
1695 SA = IMADR(N) + 56
1700 FOR I = 1 TO H
1720 EA = SA + W - 1
1730 GOSUB 1000: REM MOVE ROW
1735 SA = SA + W
1740 DA = DA + 160
1750 NEXT I
1755 GOSUB 1800: REM DO COLORS
1756 GOSUB 1900: REM DO MODE
1790 RETURN
1799 REM DO COLOR PALETTE
1800 SA = (IMADR(N) + 21)
1810 EA = SA + 32:DA = 40448
1820 GOSUB 1000: REM MOVE IT
1830 RETURN

```

```
1899 REM DO MODE/SCB'S
1900 M = PEEK (IMADR(N) + 3); IF M < 0 THEN M = 128
1910 SA = 774;EA = SA + 50;DA = 40192
1920 FOR X = SA TO EA: POKE X,M: NEXT X
1930 FOR X = 1 TO 4: GOSUB 1000
1940 DA = DA + 49: NEXT X
1950 RETURN
9999 REM ERROR TRAP
10000 IF PEEK (222) = 254 THEN 130
10005 GOSUB 1220
10010 PRINT"ERROR ";PEEK (222);" IN LINE ";PEEK(218)+PEEK(219)*256
10020 END
```

# X. The POKE That Should Be In The IIGS Control Panel

Many programs written for the Apple II+/IIe/IIc computers don't display as well as they should on the Apple IIGS, regardless of whether the monitor is RGB, composite, or monochrome.

What many people do NOT know is that the display choice (Color or Monochrome) in the IIGS control panel is for IIGS super hi-res graphics only, and not for regular hi-res of standard Apple II's. For pre-IIGS programs you should always set the display to monochrome. If you forget, you'll see only pink and green colors in some circumstances.

Pre-IIGS programs should let the user POKE either of the following two POKES:

POKE 49247,0 or POKE-16289,0 (enable color and fuzzy vertical lines)

POKE 49246,0 or POKE-16290,0 (enable B/W and sharp, clear lines)

Abracadata, for example, has done the following to get crisp, sharp lines and hi-res text characters in *Welcome* and *Everybody's Planner*: POKE 49246,0 in line 0 of the boot program if the computer was determined to be a IIGS. You should not use this POKE with pre-IIGS computers because on a IIe with an RGB monitor, POKE 49246,0 crashes programs.

You can check address 65535 (-1) to find out what kind of Apple II someone has. If it's 192, then the machine is a IIGS. This isn't the standard way to check the system type, and one possible problem is future compatibility: will future Apple machines have different values at this location? We decided to deal with that when, and if, it happens. Here's our algorithm:

0 P = PEEK (65535) : P =  $\begin{matrix} < > \\ \wedge \\ P \end{matrix}$  (P\192) : POKE 49246-P\*49240,0

You need to use 65535 rather than -1 for the address (same number, different form) to avoid an error (due to a bug in *Applesoft*). *Applesoft* doesn't like negative numbers in the first line of a program.

Color isn't needed in the two programs mentioned above, so a POKE 49246,0 for a IIGS creates a clean, sharp, monochrome display. But what about color? For example: *Architecture*, *Interiors*, and *Landscape* of our *Design Your Own Home* series, and our *Design Your Own Train* and *Run Your Own Train* products need color sometimes, and crisp, clean monochrome lines other times. In this case, and in most cases where Apple II hi-res screens will be used on a IIGS, the solution is to let the user decide when they want color and when they want sharp lines and text. So we let the users employ special commands in all five programs. They can switch back and forth between the two display modes. *We recommend this to other vendors!*



This would have been a nice display switch for the Control Panel. One wonders whether it was an oversight by Apple.

You could put the POKEs right in a program to solve the problem, setting them to automatically occur whenever the user is in a color-needing mode (49247,0), or a sharp line or text-needing mode (49246,0). Don't forget to have your program check for a IIGS machine before POKeing. A problem with this automatic method is that hi-res text and sharp lines often are found on the same screen at the same time as colored lines, or color-filled shapes and scenes, so giving the user a choice is still the best answer.

# XI. Source Code Disk Description

*Graphics Supermarket* is written the C programming language using the Apple Programmer's Workshop (APW). The source code disk contains all the source files used to create the executable version of the *Graphics Supermarket*.

The files on the source code disk are:

MAIN.C	BUFFER.C	EDIT.C
LINE.C	MENU.C	COLOR.C
DISKIO.C	GS.IMAGES.C	RECTOVALRC.C
FATPIX.C	PRINT.C	FILL.C
HELP.C	DISKIO2.C	ANIMATE.C
INIT.C	MISC.ASM	GS.A

The MAIN.C file contains the *event loop* and the *includes* for all of the remaining files. The INIT.C file has global variable declarations and tool start-up routines.

You need at least 1 megabyte of memory and two 3.5 inch drives to compile the source code. Following is a description of how to compile.

Boot APW, then put your APWC compiler in drive 1 and the GS.SOURCE disk in drive 2. Be sure to use a backup of your original source code disk. Later, you will need to delete a source file to make room for the load file created during linking. Now type...

```
prefix /gs.source [Return]
```

to set the current prefix to the GS.SOURCE volume (if you named the backup volume differently, use the new name you gave it in place of GS.SOURCE) and type the following to begin the compile.

```
compile main.c keep=gs [Return]
```

The source code is compiled into an object file named GS.ROOT. Be prepared for a wait, it takes approximately 18 minutes to complete the compile.

When the object file is created, you need to link it with the GS.A file to create the final load file. Before beginning the link, you need to delete a source file to free enough disk space for the load file. So, to delete a file and create an empty folder named TEMP, type the following.

delete buffer.c [Return]

create temp [Return]

The folder serves as a directory place holder for your deleted file name. After linking, you can delete the TEMP and GS.ROOT files, then copy BUFFER.C back to your working source disk. To start the link and create a load file named GS, type this:

line 2/start gs keep=gs [Return]

When the link is finished, you can change the filetype of the GS load file to S16 and execute it from the FINDER, or you can leave the filetype as it is and type:

GS [Return]

to execute it from APW. However, the proper folders (with drivers, tools, etc.) must be in the SYSTEM subdirectory of your APW boot disk to boot GS from within APW. (Refer to your program disk to see what files are needed in the SYSTEM folder). You can change the filetype by typing the following:

filetype gs s16 [Return]

The object file, GS.A, was created by compiling the assembly language source file, MISC.ASM. It contains mostly menu and cursor data. You can assemble the MISC.ASM file by typing (of course, you need to use your APWA assembler disk in drive 1, not the APWC disk)...

delete gs.a [Return]

to remove the current MISC.ASM object file, GS.A. Then type:

compile misc.asm. keep=g [Return]

to begin the compile. After the compile is finished, change the object filename to GS.A by typing:

rename g.root gs.a [Return]

so it will automatically link with the C object file when you compile and link the C source as described above.

We think you'll find the *Graphics Supermarket* source code to be a helpful learning tool for you. Remember though, the program AND its source code is copyrighted material, and we've provided it to you for educational purposes only. You may compile it, edit it, compile and re-compile it for personal use, but you may not sell it or use it for commercial purposes in any form.

# Index

animate 23  
arcs 37

Background Color 10, 29, 30, 41, 42, 64, 65, 72  
binary

file formats 63, 69-72  
see load, save

blend 15, 48

circle - see ovals

click 8

clipboard

animate 23  
copy 22  
cut 21, 32  
Hold 16, 19-21  
Display 17, 20  
menu 15 - 23, 51-53  
memory usage 24-27  
Paste 16, 20-22  
Release 17, 22  
see load, save

Close 50, 52

color

Background Color 10, 29, 30, 41, 42, 64, 65, 72  
blending 15, 48  
changing 13-15, 39-48  
copying 13, 44-48  
default 13, 43, 48, 52  
dithering 39, 40, 43, 47  
Draw Color 10, 13, 41  
icons 39, 41  
palette 14, 39-48  
mode 39-45  
RGB 14, 39, 43-44  
table 39, 40, 43, 47, 64, 66, 70  
swapping 14, 44, 45

Control Panel 8, 54, 55, 87

copy

image 22  
colors 13, 44-48

cut 21, 32

- delete 51, 53
- dialog box
  - buttons 12, 49
  - description 11, 49
  - disk 49-52
  - edit fields 12, 49
  - options 12, 49
  - printer 56-62
  - scroll bar 11
- Display 17, 20
- Display area 17, 20
- disk (see load, save)
- dithering 39, 40
- drag 8
- Draw color 10, 13, 41
- drawing 13, 29 (also see color)

- Edit Area
  - description 9
  - flipping 32
  - scrolling 18, 33
  - size 10, 16, 18, 21, 22, 24, 30, 31
  - special use 21, 22, 47

- edit field 12

- Edit Frame
  - description 10
  - dragging 30, 31
  - hide - show 32
  - moving 32
  - size 10, 16, 18, 21, 22, 24, 30, 31
  - special use 21, 22

- erase

- Display Area 16, 30
- Edit Area 16, 29, 32
- screen 13, 17, 29, 30, 36

- file formats 63, 66 (also see load, save)

- file type 50, 52, 69 (also see file)

- fill 37

- Hold 16, 19-21

- Icon Bar (see icons)

- icons 9, 34, 39, 41

- Image Size 10, 16, 18, 21, 22, 24, 30, 31

- invert 62

- items 9

lines 13, 37

load

binary 52

Clipboard 16, 17, 51, 52

disk 15-17 51, 52

pictures 52

source code 52, 53

Magnify 34-36

mask 64-66, 72, 74

memory 7, 24-27, 54

Menu Bar 9

menu items 9

menus

CLIPBOARD 15-18, 19-23, 51-53

COLOR 39, 41, 43, 48, 52

color change effects 47

DRAW 34-38

EDIT 29-43

FILE 49-53

MODE 28

PENSIZE 13, 34

Selecting 9

STATUS 24-27

modes 17, 28, 39 (also see color)

mouse 8

Open 50, 52

ovals 37

palette 14, 39-48

Paste 16, 20-22

Pensize 13, 34

pixels 11, 28, 34-36, 44-47

printing

cards 17, 18

images 54-62

programming images 63-86

pull-downs 9

ram (see memory)

rectangles 37

Release 17, 22

RGB 14, 39, 43-44

Save

binary 50, 63, 65, 69-72

Clipboard 19, 50, 63-65, 70

disk 16, 49, 63

pictures 50, 69

source code 50, 51, 63-65

Scroll 18 (also see Edit Area)

Scroll Bar 11 (also see dialog box)

sketch 13, 36

source code

disk 89 (also see load, save)

file formats 63, 66-69, 72-80

spray 36

status 24-27

swapping colors 44, 45

update screen 42

Undo 13, 29, 33, 43, 46, 47

## LIMITED WARRANTY, DISCLAIMER OF WARRANTIES, AND LIMITATIONS OF LIABILITIES

Abracadata, Ltd. warrants to the licensee and no other person or entity the disks on which GRAPHICS SUPERMARKET are recorded (but not the software nor this manual) to be free from defects in materials and workmanship under normal use for a period of 30 days after the date of original purchase. If during this 20-day period a defect should occur, the disk may be returned with a copy of your dated sales receipt and a brief description of the problem to Abracadata, Ltd. P.O. Box 2440, Eugene, OR 97402, for replacement free of charge. Your sole remedy in the event of a defect in a disk is limited to replacement of the disk as provided above. Any implied warranties applicable to the disk shall terminate not later than 30 days after the original date of purchase. If the disk has been physically damaged, is beyond the 30 day limit, or is not accompanied by proof of purchase, include \$10 per each disk plus \$3.55 postage and handling per request. Please return the disk only, not the entire package. Do not staple or paper clip the disk. Package the disk carefully to prevent bending, and mark the package "Do not X-Ray or Use Magnets." Ship via UPS or U.S. Postal Service with return receipt requested. Abracadata, Ltd. is not responsible for disks not received when you cannot provide proof of your shipment and our receipt.

GRAPHICS SUPERMARKET is sold or licensed on an "as is" basis. Except as provided above with respect to the disks, Abracadata, Ltd. makes no other warranty or guarantee of any kind, express or implied or statutory, with respect to GRAPHICS SUPERMARKET, its quality, performance, accuracy, merchantability, or fitness for any particular purpose; and assumes no responsibility or liability to the licensee or any other person or entity with respect to any damage caused or alleged to be caused by use of GRAPHICS SUPERMARKET. The entire risk as to the quality, suitability, accuracy and performance of GRAPHICS SUPERMARKET is with the licensee. Except as provided above with respect to the disk, if GRAPHICS SUPERMARKET proves defective after purchase, the licensee (and not Abracadata, Ltd., its distributor, its retail seller or others involved in the development of the product) assumes the entire cost of all necessary servicing, repair, or correction. In no event will Abracadata, Ltd. be liable for direct, indirect, incidental or consequential damages resulting from any defect in GRAPHICS SUPERMARKET or from any particular application of GRAPHICS SUPERMARKET by the licensee or any other person or entity, even if Abracadata, Ltd. has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. If any exclusion or limitation shall be determined by a court to be invalid or enforceable, the determination shall not affect in any respect the validity or enforceability of remaining exclusions or limitations or of the same exclusion or limitation in different circumstances.

Abracadata Ltd gives no assurance that the information used in the development of GRAPHICS SUPERMARKET is reliable and accurate. The recommendations by GRAPHICS SUPERMARKET represent only the opinion and judgement of the author and are intended to be applicable in normal circumstances. Individual circumstances may vary and may render certain recommendations inapplicable. AL gives no assurance that the opinions of judgement of the author are reliable and accurate.