Interactive Image Author

An Authoring Tool for Creating Interactive Graphic Files

by

Matthew J. Zukoski

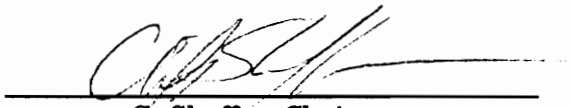
Project Report submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

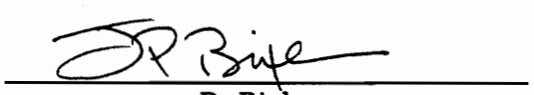
in

Computer Science

APPROVED:



C. Shaffer, Chairman



P. Bixler

J. Burton

August, 1990
Blacksburg, Virginia

c.2

LD
5655
V851
1990

Z866
C.2

Interactive Image Author

An Authoring Tool for Creating Interactive Graphic Files

by
Matthew J. Zukoski

Committee Chairman: Cliff Shaffer
Computer Science

(ABSTRACT)

The power of the computer has yet to be utilized to its fullest with regards to teaching. In particular, the medical field continues to use teaching strategies such as the teaching file, overhead transparencies, 35mm slides, printed quizzes and a pointing stick.

Today, powerful personal computers offer high-resolution graphic capabilities and sophisticated development environments that make it possible to create computer-assisted education environments for studying medicine. The Interactive Image Author is an authoring tool for creating interactive graphic files that can be used to develop computer-based teaching files and quizzes on the Macintosh computer. The Interactive Image Author demonstrates how the microcomputer can provide a new personalized teaching approach for doctors and teachers in other fields as well.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Cliff Shaffer, for his continued support throughout this entire effort. Without his encouragement, this project would not have become a reality. His assistance has been greatly appreciated. In addition, I would like to extend thanks to the other members of my committee: Dr. Pat Bixler and Dr. John Burton. They have been of great assistance throughout the development of this project.

Special thanks to the sponsors : The Radiology Department of the Milton S. Hershey Medical Center, Dr. J. S. Mayer, and Dr. J. Madewell.

Apple, HyperCard, HyperTalk, Macintosh, and MultiFinder are trademarks of Apple Computer, Inc.

LightSpeed C is a trademark of Think Technologies, Incorporated

The MacRecorder is a product of Farallon Computing, Incorporated

Exposure is a trademark of Preferred Publishers, Incorporated

CONTENTS

AN INTRODUCTION TO THE PROBLEM.....1

Characterizes the type of learning problem surrounding computer-assisted instruction and explains how interactive, graphic-based software can help.

THE INTERACTIVE IMAGE AUTHOR PROTOTYPE PROJECT.....5

A brief history of the Interactive Image Author project.

MACINTOSH AND C.....8

Lists some basic reasons behind the choice of using the Macintosh as the platform and why C was chosen as the language in which the system would be written.

SYSTEM DESIGN.....11

Describes many of the complications encountered while developing the system and how these problems were handled on the Macintosh.

THE SYSTEM IN USE.....20

Discusses how the Interactive Image Author system is being put to use at the Hershey Medical Center.

FUTURE PLANS FOR INTERACTIVE IMAGE AUTHOR.....21

Discusses how the Interactive Image Author system is being planned for demonstration at upcoming conferences and how it will be put to use in areas other than medicine.

REFERENCES.....23

APPENDIX A: *INTERACTIVE IMAGE AUTHOR MANUAL*

APPENDIX B: *INTERACTIVE IMAGE FILE SPECS*

AN INTRODUCTION TO THE PROBLEM

A doctor, medical student, or student in any other field today has many opportunities for continuing education. There are seminars, conferences, review books and journals. Each has limitations and deficiencies. One key problem with these methods is their inability to tailor education to the individual. Every student has special needs: some information presented may already be known, while other information of interest may not be treated in sufficient detail. In the medical field, doctors continue to use dated teaching strategies such as the teaching file, overhead transparencies, 35mm slides, printed quizzes and a pointing stick. In all fields, the power of the computer has yet to be utilized to its fullest as a teaching tool.

Currently there are powerful yet affordable personal computers that offer high-resolution graphic capabilities and sophisticated development environments that make it possible to create computer-assisted education environments for studying medicine, geography, chemistry, or any other field.

Most computer-assisted instruction (CAI) programs focus primarily on *computing* that assist in basic computational skills (such as adding and

subtracting). Only recently has there been an increase in the availability of software dealing with social studies and language arts. While the number of programs dealing with social studies and language arts have been increasing, the number of new science programs have decreased (see Figure 1) to the disappointment of many educational technologists who are concerned about the state of science teaching in the United States. [1]

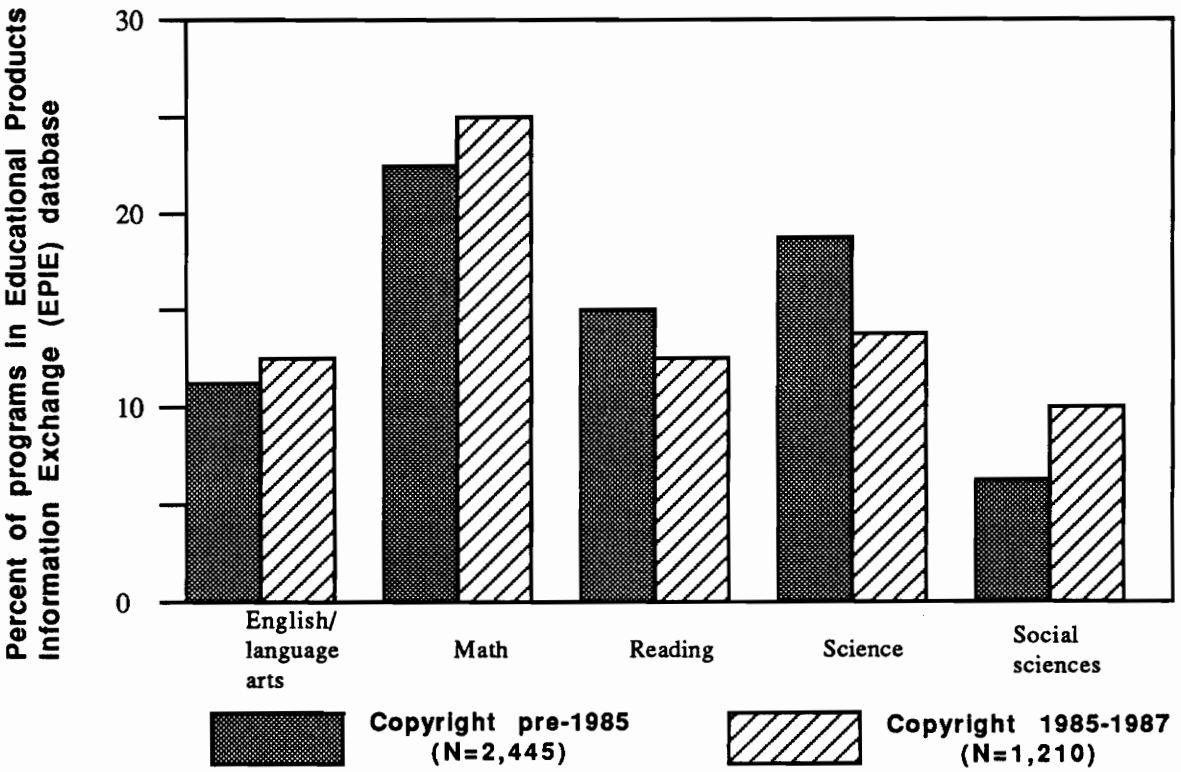


Figure 1 - Trends in Availability of Software for Major Subject Areas

CAI can play an important role in all sciences, including medicine.

Chen, Hoffer, and Sweet have found that CAI in radiology "has been shown to be as effective as conventional lecture formats, and subjectively rated superior by students because of its interactive nature." [2]

EDUCATION IN RADIOLOGY

While teachers in all fields confront difficulties in staying current, the medical community faces some additional obstacles in their field. In particular, radiology relies on new technology to provide new and more effective solutions to medical problems. For example, computed tomography, magnetic resonance, and ultrasound are some of the new imaging technologies used in radiology. Radiology is a rapidly expanding field. This expansion is making it difficult for a radiologist to remain current. It has caused them to become increasingly specialized. Yet, many are expected to be both experts in their area of specialization and capable of handling a wide array of general disorders. Many of these diseases are extremely rare, adding to the radiologist's problem of gaining experience.

Radiology departments have been confronting this problem by accumulating special case studies of various patients and constructing *teaching files*. These files contain information pertinent to the patient's condition that will allow the medical student to accurately diagnose the disease. The student reviews the information presented in the file and arrives at a diagnosis. Their diagnosis is checked against the correct

diagnosis on the back of the file.

The majority of medical schools still have not incorporated the computer as an effective tool for teaching. An appropriate computer-assisted instruction environment would allow doctors or teachers in the medical school to develop teaching files and quizzes and provide more advantages over non-computer teaching methods.

Today, the technology exists to incorporate detailed images, sound and animation into a computer-assisted lesson. This includes high-resolution X-rays, CT scans, ultrasound scans, the actual sound digitization of a heart beat, and an animated sequence of the blood flowing through the chambers of the heart. This is especially important to radiologists who rely on many forms of information to reach a diagnosis.

THE INTERACTIVE IMAGE AUTHOR PROTOTYPE PROJECT

Early in the Fall of 1988, the Radiology Department of the Milton S. Hershey Medical Center in Hershey, Pennsylvania, expressed an interest in designing computer-assisted-instruction (CAI) tools for doctors and residents of the hospital. The hospital envisioned tools that would be capable of presenting medical students with the information and experience they typically gained in the classroom, but at a much more interactive level. In addition, doctors engaged in research would be able to utilize the tools as time-saving learning and teaching devices. Initially, these tools would be geared toward radiologists, but they should be designed with wide-ranging applications as the ultimate objective.

Drs. John S. Mayer and John Madewell identified four key aspects in the education process at the Hershey Medical Center:

Basic Teaching - Doctors and residents attend seminars and lectures on various topics.

Case Studies - Residents and doctors use detailed teaching files to learn basics in diagnosis and increase their exposure to different areas in radiology.

Tests - As could be expected, quizzes and tests play an important role in the learning environment, especially for residents.

Library - The Medical Center has an extensive research library. Teaching files and lectures often refer to various works in the field of radiology, and residents are expected to access this large body of information often.

Drs. Mayer and Madewell wanted to improve the current learning environment at the Medical Center by making information more accessible, highly interactive, and enjoyable. The new educational systems would have to be capable of emulating current education techniques as well as provide new possibilities for learning. After careful consideration of the goals of the medical center, it was decided that the *interAct!* system would be developed. It is a system composed of five basic parts: Lecture, Teaching File, Glossary, Reference, and Quiz. In addition, it was decided that an authoring tool be developed to allow doctors to easily create new topics for study. A model of the *interAct!* system was developed by Darren Dittrich and is described in [3]. The *interAct!* system is able to:

- Present a short lecture on several topics in textual form with links, associated graphics, and animation sequences.

- Display graphics: both bit-mapped graphics and high quality gray-scale images accurately displaying radiographs.

- Keep a glossary of important terms and definitions that are linked to the lecture.

- Provide for references and bibliographies that are linked to the lecture and quiz.

• Provide a quiz section capable of presenting the user with True/False, Multiple Choice, and Matching questions on the subject presented.

• Provide a teaching file which is capable of presenting the doctor with a patient history and accompanying radiographs and elicit a diagnosis. [3]

Each of these subsystems have been integrated to form one single interactive tool. The system was designed to be a model and was developed using HyperCard. Because of this reason, the system has limitations. In particular, the quiz and interactive graphic screens were *hard-coded* (or fixed). They are not easy to modify and additional quizzes or interactive images require the expertise of a HyperCard programmer. With this in mind, the Department discussed the development of new tools that would enable any resident or doctor to create interactive images and quizzes. Interactive Image Author was designed and developed to create these interactive files.

MACINTOSH AND C

The Department of Radiology at the Hershey Medical Center has made a commitment to computer technology. Their goal is to provide a computer for every secretary's and doctor's desk in the department. A committee was established to choose a computer that would be set as the standard for the department. Up until that time, there had been several Apple II and IBM personal computers in the department. After careful consideration, the Macintosh computer was chosen for its ease of use. It would minimize training for doctors and secretaries. The department currently owns thirty-two Macintosh computers. They include Macintosh SE's, Macintosh SE/30's, Macintosh II's, Macintosh IIcx's, and Macintosh IIci's.

In the Summer of 1989, the department established a research section that would be responsible for computer support for the department and investigation of computer-based education in medicine.

Since the department made the decision of the Macintosh as the computer of choice, the research software would have to be developed on the Macintosh in as much as it would be tested by the doctors and residents in the department.

The research group designed a hypertext-like system that would be highly interactive and easy to use. The initial prototype, interAct!, was

developed in HyperCard.

Because of the limitations of HyperCard, it was found to be difficult to expand the capabilities of the system without writing numerous XCMD's. An XCMD is a sub-program written in a high-level language such as Pascal or C that can be called by HyperCard. XCMD's allow HyperCard programmers to overcome the limitations of HyperCard. It was felt among the research group at Hershey that many XCMD's would be required if the Interactive Image project were to be done in HyperCard. For example, HyperCard is limited to rectangular regions. To create an irregularly shaped region required approximating the region with many rectangles. XCMD's would be needed to handle these regions. XCMD's would also be needed to handle PICT files in multiple windows. Also, HyperCard is very slow since it is an interpreter. It is memory hungry and only supports black and white images. While XCMD's could be written to get around these deficiencies, the group felt that if too many XCMD's would have to be written to supplement the HyperCard program, it made more sense to write the entire system in a high-level language. For this reason, the decision was made to do all future development in C.

Currently, there are two major platforms for programming in C on the Macintosh. They are Macintosh Programmers Workshop (MPW) and LightSpeed C. MPW is a huge system which is centered on the MPW shell. The editor and compilers are executed through the shell. It is designed for very large projects and it has a slow edit-compile-execute cycle. By

contrast, the LightSpeed C environment is designed only for C projects. It's edit-compile-execute cycle is very fast. LightSpeed C was chosen as the development programming environment for Interactive Image Author.

SYSTEM DESIGN

Event-driven Programming

Macintosh programs, such as Interactive Image Author, are based on a main event loop. That is, Macintosh programs are *event-driven*. **Events** are the part of the Macintosh Toolbox that allow the application to monitor the user's actions, such as those involving the mouse and keyboard. Whenever the user presses a mouse button or types on the keyboard, the application is notified by means of an event. The application decides what to do from moment to moment by asking the Event Manager for events and responding to them one by one in whatever way is appropriate.

When an event takes place, a check is made in the main event loop to determine what event took place. For example, when a **mouseDown** event takes place, the application must determine *where* the mouseDown took place. Possible locations are in the menu bar area, in a close box of a window, in the grow region of a window, in the drag area of a window, or in the content area of a window. When Interactive Image Author gets a mouseDown in a window, it finds out which window was clicked in using the **FindWindow** function. A check is then made to determine which window was clicked. It could have been the Tools Palette or an image window.

If a click was made in the Tools Palette, the location of the mouse is

determined relative to the Tools Palette window. From this mouse location, the application can determine which tool was clicked. That tool is set to be the current tool.

If the click was made in the image area, the action taken is determined by what the current tool is.

If the current tool is the arrow, a click in the image area causes the application to *loop through* the entire region list. If the mouse location is found to be in one of the regions, the user can *drag* the region by holding the mouse button down and moving the region to a new location before releasing the mouse button.

Regions

Graphics are an important part of every Macintosh application. All graphic operations on the Macintosh are performed by **QuickDraw**. QuickDraw is a set of graphic routines found in the Macintosh ROM Toolbox. The Toolbox is a common set of routines that every Macintosh programmer can access. The Toolbox not only ensures familiarity and consistency for the programmer and user, but also reduces the application's code size and development time.

Unlike most graphic packages that can manipulate only simple geometric structures (usually rectilinear), QuickDraw has the ability to gather an arbitrary set of spatial coherent points into a structure called a **region**, and perform complex yet rapid manipulations and calculations on

such structures. The key feature to the interactive image files created by Interactive Image Author are these regions.

A region is defined by calling routines that draw lines and shapes. The outline of a region should be one or more closed loops. A region can be concave or convex. Regions can consist of one area or many disjoint areas. They can even have "holes" in the middle. In Figure 2, the region on the left has a hole in the middle, and the region on the right consists of two disjoint areas.

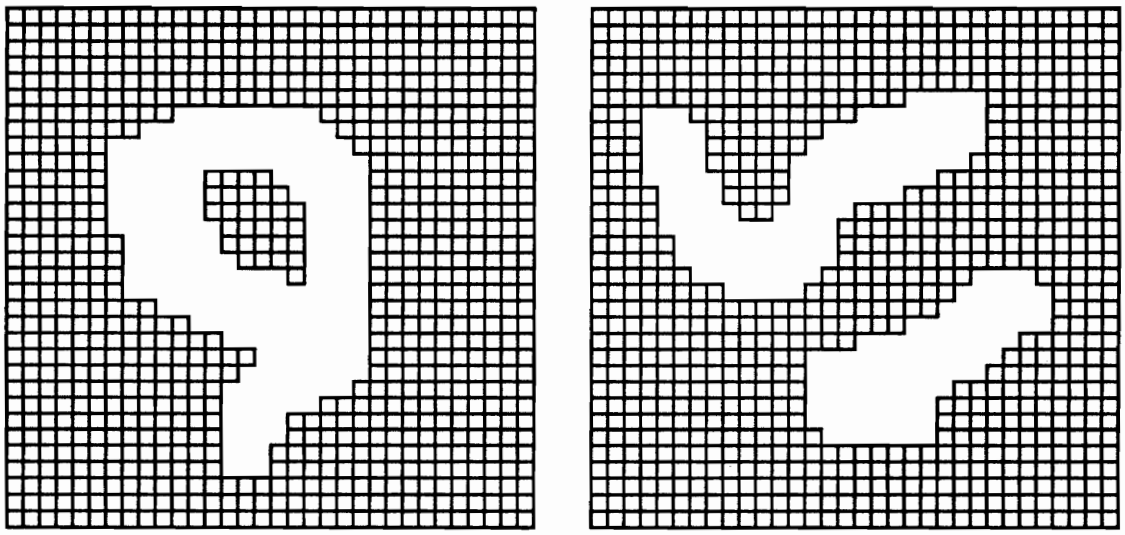


Figure 2 - Regions

Many calculations can be performed on regions. For example, a region can be *expanded* or *shrunk* and, given any two regions, QuickDraw can find their union, intersection, difference, and exclusive-or. It can also determine whether a given point intersects a region, and so on.

Macintosh regions are used extensively by Interactive Image Author.

Three tools are available for creating regions. The lasso tool is used to create free-form regions. The box tool is used to create rectangular regions and the polygon tool is used to create polygon regions (free-form with straight edges).

The "point intersecting a region" routine, **PtInRgn**, was used to determine if a mouse click by the user in the image area was inside of a region. If so, it checked for a *double-click* (as a shortcut to getting info for the region) in the region or a *drag* in the region (to move the region).

Creating Regions

If the current tool is the lasso, a click in the image area causes Interactive Image Author to open a new region using the **NewRgn** function. As the user holds the mouse button down, the points are added to the region definition. When the user releases the mouse button, the region definition is closed using the **NewRgn** function. The region is then added to the region list, and the region is drawn on the image. Since regions must be closed, if the user releases the button before returning to the starting location, a line is drawn from the last point recorded to the initial point. This guarantees a closed region. The algorithm for creating a lasso region looks like:

```
rgnHdl := NewRgn();
startPt := starting location
OpenRgn();  { Start collecting points }
      MoveTo(startPt.h, startPt.v);
      WHILE Button() THEN BEGIN
            GetMouse(where);
            LineTo(where.h, where.v);
      END

      IF (NOT EqualPt(startPt, where)) THEN
            LineTo(startPt.h, startPt.v);
CloseRgn(rgnHdl);
```

If the current tool is the box, a click in the image area causes the application to open a new region. The point is recorded as one of the corners of the box. After the user drags to the corner opposite of the initial corner and releases the mouse button, the region is closed and is equal to the box defined by those two opposite corners. The region is added to the region list and the region is drawn on the image. The algorithm for creating a box region looks like:

```
rgnHdl := NewRgn();
startPt := starting location
WHILE Button() THEN
  ; { Do nothing until opposite corner is selected }
GetMouse(endPt);
Pt2Rect(startPt, endPt, theRect);
OpenRgn();
      FrameRect(theRect);
CloseRgn(rgnHdl);
```

Creating the polygon region was not as easy as the lasso and box regions. When the lasso region is being created, the MoveTo and LineTo procedures are called while the region is open. When a region definition is open, all subsequent drawing commands to the screen affect the region definition. When a polygon region is being created the user is allowed to move the mouse freely before he clicks the button to set each vertex. As the mouse is moved freely, the MoveTo and LineTo procedures are used to erase the previous line and to draw the new line. It is actually animation. The mouse locations of each mouse click need to be stored. With this is mind, a linked list structure was used to hold the vertices as the user created the polygon region. As the user continually points and clicks (forming line segments), the endpoint of each segment is added to a linked list. When the user double-clicks the mouse button or returns to the initial point, a new region definition is created using NewRgn and the following algorithm is used to create the polygon region:

```
IF (vertexList <> nil) THEN BEGIN
      currNode := vertexList;
      currVertex := currNode^.theVertex;
      MoveTo(currVertex.h, currVertex.v);
      currNode := currNode^.next;
      rgnHdl := NewRgn();
      OpenRgn();
            WHILE (currNode <> nil) DO BEGIN
                  currPoint := currNode^.theVertex;
                  LineTo(currVertex.h, currVertex.v);
                  currNode = currNode^.next;
            END;
      CloseRgn(rgnHdl);
      disposeOfVertexList(vertexList);
END;
```

If the current tool is the union tool, a click in the image area causes the application to get a region created by the user as they would by using the lasso. A union operation is performed between this newly created region and the *current region*. This union operation is implemented in the ToolBox found in the QuickDraw unit called **UnionRgn**. The algorithm for performing a union on the current region is:

```
currRgnHdl := regionList[currRgn];
getLassoRgn(theRgnHdl);
UnionRgn(currRgnHdl, theRgnHdl, currRgnHdl);
```

If the current tool is the intersect tool, a click in the image area

causes the application to get a region created by the user as they would by using the lasso. An intersection operation is performed between this newly created region and the *current region*. This intersect operation is implemented in the ToolBox found in the QuickDraw unit called **SectRgn**. The algorithm for performing a union on the current region is:

```
currRgnHdl := regionList[currRgn];
getLassoRgn(theRgnHdl);
SectRgn(currRgnHdl, theRgnHdl, currRgnHdl);
```

When creating regions for an interactive image, the border of each region may be obvious. However, it is not always clear what form the region should take. There are times when you may want to extend the region's border to make it easier for the user to select regions in the quiz program. For example, suppose you are creating interactive images that will be incorporated into a quiz for second graders. If you want to ask them a question to find a frog in the image, you may want to create a region that encompasses the frog and the area around the frog, for example a quarter inch away. Children may have a difficult time hitting exactly on an object. It is similar to the situation of children coloring in a coloring book. They do not always color inside the borders. On the other hand, if you are creating a teaching file for medical students, you probably want to outline the regions pixel-by-pixel, so they have to be exact in finding a tumor or lesion.

Sound

Interactive Image Author makes use of resource type 'snd ' sounds. The 'snd ' type is the standard sound type used on the Macintosh today (analogous to the PICT type being the standard graphic format). The **Select Sound File** command found in the **Sound** menu will associate a sound file with the interactive image. This sound file must have resource type 'snd ' sounds in it. The **Resource Manager** calls are used to determine if this is so.

The sounds are actually played by using the **Sound Manager** procedure, **PlaySound.**

THE SYSTEM IN USE

Initially, the intent of Interactive Image Author was to be used by the medical community to create interactive image files. These files would be used to create quizzes and lessons that the doctors could give to their students. Presently, there are several lessons being developed with radiologists in the Department of Radiology. Use of the system has since spread to other areas. Lessons are currently being developed with local elementary and secondary teachers.

A chemistry lesson and quiz is being developed to instruct students on the periodic table. For example, the periodic table is presented on the screen and the student is asked to find Magnesium.

An art lesson is being developed to help secondary art students study European art from the ancient times to the present time.

A lesson has been developed to teach second grade students punctuation. It is currently being tested with a select group of children.

A geography lesson is being developed to interactively quiz students on states of the United States and countries of the world.

FUTURE PLANS FOR THE INTERACTIVE IMAGE AUTHOR

Five lessons are currently being developed by doctors in the Department of Radiology. The lessons deal with radiology physics, ultrasound, the chest ("Cross-sectional Anatomy of the Chest"), the heart ("Vascular Rings and Other Aortic Malfunctions : An Interactive Computerized Tutorial"), and brain lesions ("Angiotopography of Intracranial Lesions"). These lessons have been approved to be presented at the Radiological Society of North America (R.S.N.A.) conference in Chicago, Illinois in November, 1990. The Radiological Society of North America is one of the largest medical societies in the world. Every year, the R.S.N.A. conference is attended by more than 40,000 people to witness new discoveries and inventions in radiology. Five stations will be set up for each of the five lessons. Valuable feedback will be gathered from attendees who experiment with the lessons.

Following the conference, the lessons will be used by the medical students and residents in the Department of Radiology at the Hershey Medical Center. It has not been determined at this time if the interactive Macintosh quizzes will be mandatory in the Spring semester of 1991 or if the quizzes will be given to a select group of volunteers. The chairman of the department is discussing the possibilities with the faculty and a decision

will be made by the end of 1990.

The lessons being developed with local elementary and secondary teachers will be utilized and tested by their students during the 1990-1991 school year. Data from these experiments will be collected and analyzed.

REFERENCES

1. Educational Products Information Exchange, U.S. Congress, Office of Technology Assessment, Washington D.C., July 1987

2. C. Chen, P. Hoffer, and H. Sweet, Hypermedia in Radiology, Yale-New Haven Hospital, New Haven, CT, 1989

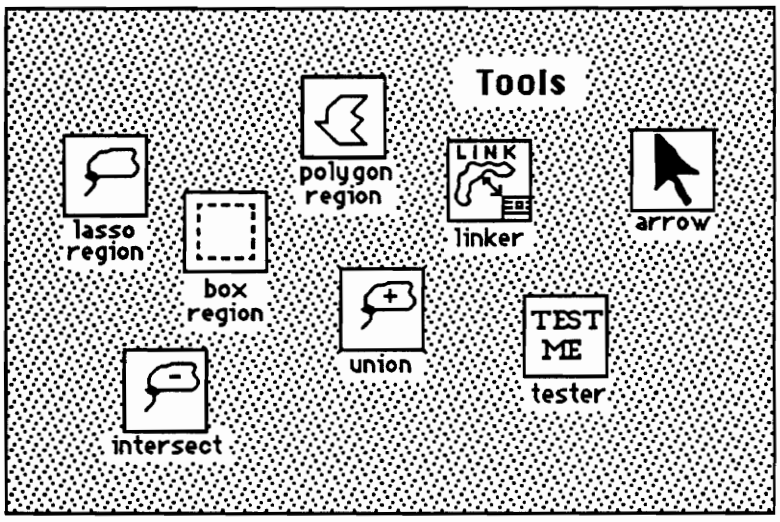
3. Darren Dittrich, *interAct!* - A Multimedia Educational System for Medical Professionals, Project Report, Virginia Polytechnic Institute and State University, September, 1989, pp. 5-7

4. Inside Macintosh Volumes I-V, Apple Co., Inc., 1985-1988

Appendix A

The Interactive Image Author Manual

Interactive Image Author Users Manual



Contents

Preface **Getting started**

What's on your Interactive Image Author program disk 4

About the manual 4

Chapter 1 **Quick Tour of Interactive Image Author**

Starting up 5

Loading a picture 7

Creating some "regions" 14

Saving your "interactive image" 27

Chapter 2 **The Interactive Image Author**

What is an Interactive Image? 29

The use of monochrome, gray-scale and color 30

Chapter 3 **Regions**

Creating regions 32

The lasso 34

The box tool 35

The polygon tool 37

Editing regions 38

Using the intersect tool 39

Using the union tool 41

Deleting a region 42

Moving a region 43

The use of pen color 44

Chapter 4 Labels and Descriptions

Assigning labels and descriptions to regions 47

Modifying the text of labels and descriptions 49

Re-sizing and moving the description boxes 50

Chapter 5 Links

Creating a link 54

Modifying a link 55

Deleting a link 55

Chapter 6 Working with Sound

Creating sounds 57

Associating sound to a region 59

Chapter 7 File Management

Saving an interactive image file 62

Opening an interactive image file 63

Working with multiple interactive image files 65

Preface

Getting Started

Before you start using Interactive Image Author, you should make a backup copy of the disk. The software is unprotected.

What's on your Interactive Image Author disk

The *Interactive Image Author* disk contains the following:

- The Interactive Image Author program
- System Folder containing System 6.04
- sample interactive image documents

About the manual

This manual is divided into 6 chapters:

- Chapter 1 offers a quick start for those who don't like to read manuals and want to get started right away

- Chapter 2 gives a general overview of the Interactive Image Author

- Chapter 3 describes how to create and edit regions

- Chapters 4, 5 and 6 show how to associate text and sound with each of the regions

- Chapter 7 discusses interactive image file management

Chapter 1

Quick Start

This chapter explains the fundamentals of Interactive Image Author. It is intended for those who are anxious to begin building some interactive image files quickly.

Starting up Interactive Image Author

1a. *Mac users with a hard drive:*

i. Insert the program disk into the floppy drive
ii. Drag the floppy disk icon onto the hard drive icon
iii. A message will be given stating that a new folder will be created, etc... Click **OK**.
iv. Eject the floppy and put it away in a safe place.
v. Open the newly created folder.
vi. Proceed to step 2 below.

1b. *Mac users without a hard drive:*

i. Insert the program disk into the floppy dive
ii. Open the disk by clicking on the floppy disk icon and selecting **Open** from the **File** menu (or simply double-click on the floppy disk icon).
iii. Proceed to step 2 below.

2. Open the Interactive Image Author application by clicking on it and selecting **Open** from the **File** menu (or simply double-click on it). The *About box* will appear as shown below:

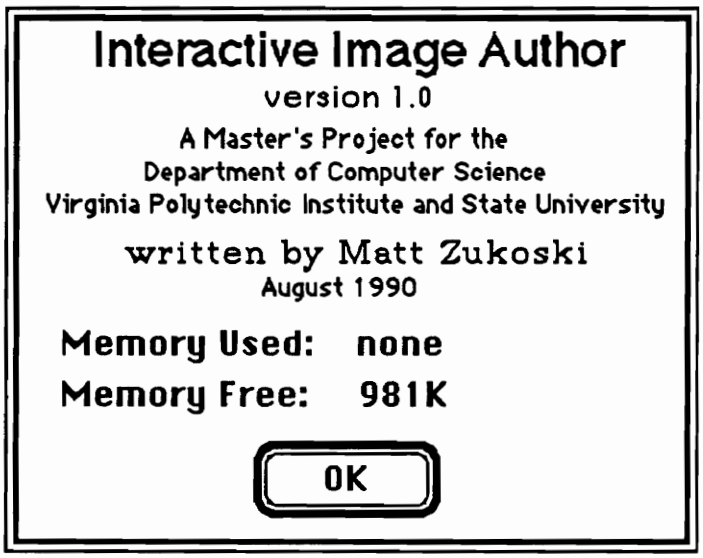


Figure 1.1 The About box

This dialog gives information about the program name, version number, who wrote the software and how much memory is available.

Click **OK** to go on. *(Note: the About box can be brought up again by selecting **About Interactive Image Author** from the menu).*

Opening a file

Choose **Open** from the **File** menu. A dialog box will appear as shown below. *(Note: the exact listing of files shown here will probably be different from the listing shown on your screen).*

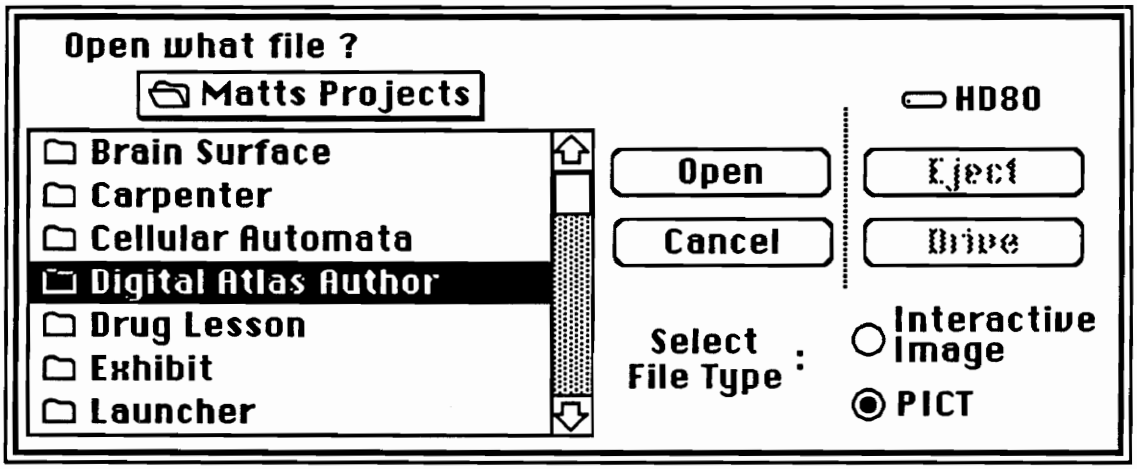


Figure 1.2 The Open Dialog Box

Locate and open the folder called **Sample Files** given to you on the Interactive Image Author disk. If you have a monochrome monitor, open the file called **B/W Stick Man** in this folder. If you have a color monitor, open the file called **Color Stick Man** in this folder. The interactive image file will be read, a little tune will play and the Tools palette will appear along with the interactive image file (see Figure 1.3).

The tools palette will be used to select the tools we need to create and edit our regions. We will select a tool and use it to operate on our image. Occasionally we will need to pull down a menu from the menu bar which appears on the top row of the screen. There are six menus: **, File, Edit, Region, Sound** and **TextFields**. You will see the use of these menus as we go along. To start off, let's take a look at the tools palette.

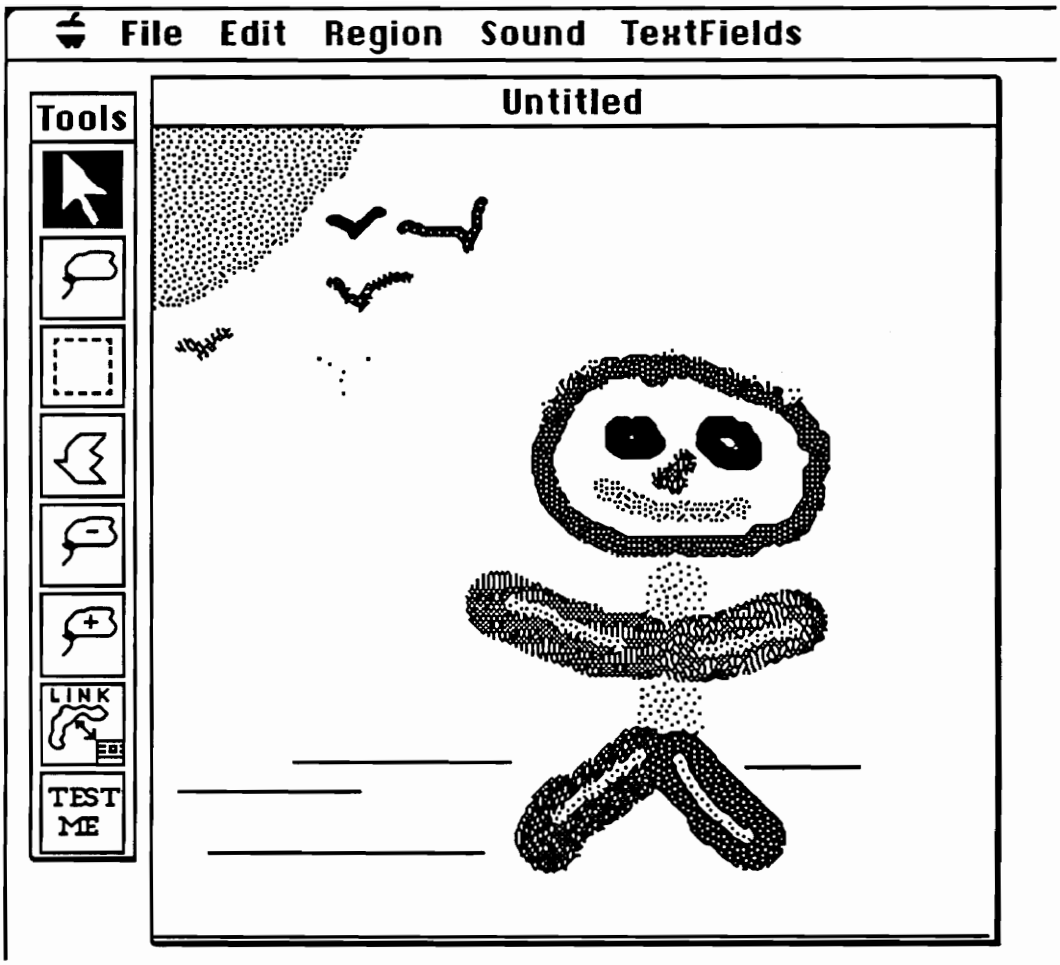


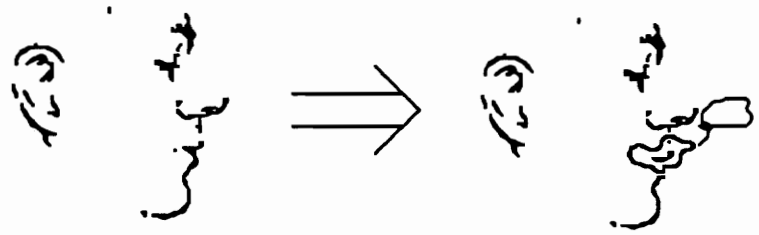
Figure 1.3 The Stick Man interactive image

The Tools Palette

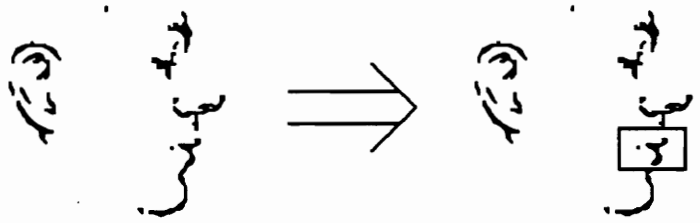
There are eight tools in the palette.

The **arrow tool** is used for selecting (pointing to) regions so that they may be *edited*, *deleted* or *moved*.

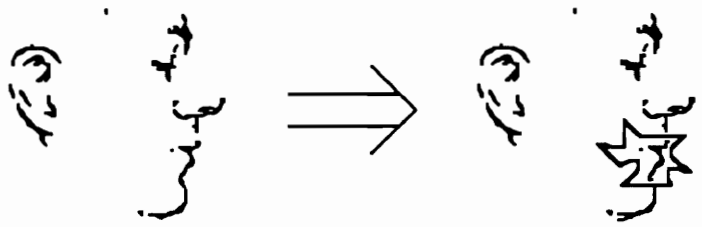
The **lasso tool** is used to create a free-form region. That is, the region can be any shape at all, oval, square, etc...



The **box tool** is used to create rectangular regions.



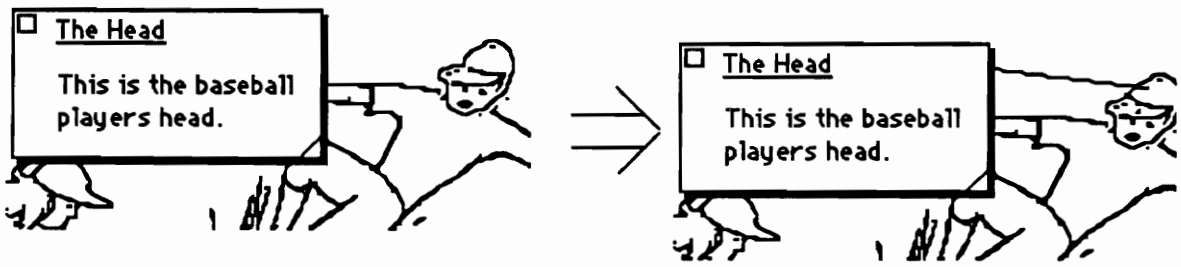
The **polygon tool** is used to create regions of any shape, similar to the lasso. However, the region is approximated by line segments.



The **intersect tool** is used to edit a region by *deleting a piece* of the region.

The **union tool** is used to edit a region by *adding a piece* to the region.

The **linker tool** is used to lay a link from the region to its corresponding description box.



The **test-me** tool is used to test the regions to check where the descriptions pop up and to hear what the voices sound like.

Selecting the Sound File

Let's use these tools to create a sample interactive image. First, choose **Select Sound File...** from the **File** menu (or press ⌘-F) to associate a sound file with our Stick Man image.

You will then be asked to choose a sound file. One has already been prepared for you. It is called **Stick Man Sounds** and it can be found in the folder, **Sample Files**. These sounds (and voices) were recorded using MacRecorder™ from Farallon Computing Inc. It's a piece of digitizing hardware that plugs into the Mac serial port. It comes with software to edit the digitized sounds.

A dialog should appear on your screen as shown in Figure 1.4. Find the file **Stick Man Sounds** and **Open** it.

Now that the Stick Man Sounds file is associated with the Stick Man image, we can associate sounds from the sound file to any of the regions that we create.

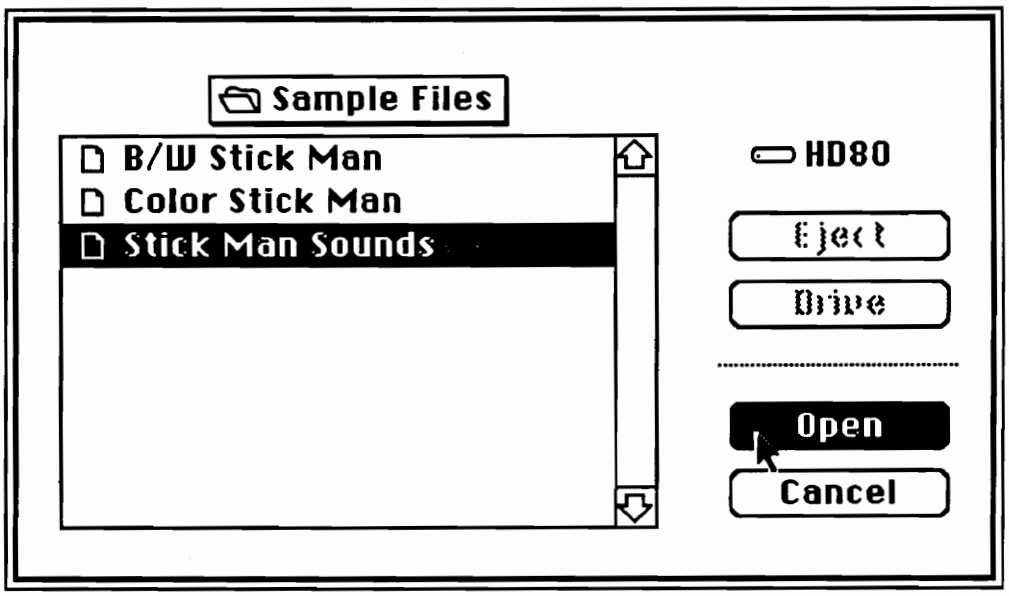


Figure 1.4 The Select Sound File Dialog

Creating Regions with the Lasso Tool

Let's create some regions. *Select* the lasso tool from the Tools Palette by clicking on it once. We're going to create a region around the right eye (that is, the Stick Man's right eye). Move the mouse over to the image. Notice how the *cursor* changes from the arrow to the lasso. Choose any point on the perimeter of the right eye. Position the tip of the lasso on that point and hold the mouse button down. As you hold the button down, outline the perimeter of the eye. Release the mouse button when you return to your starting point. See Figure 1.5.

Figure 1.5 Using the lasso tool to create a region around the eye

Note: You don't have to return EXACTLY on your starting point. In fact, you can release the button at any point. The region will close up by drawing a straight line from where you released the button to the starting point.

Entering Region Information

After releasing the mouse button, you will be presented with the dialog as shown in Figure 1.6. Type **The Right Eye** for the name. This is the label for the region.

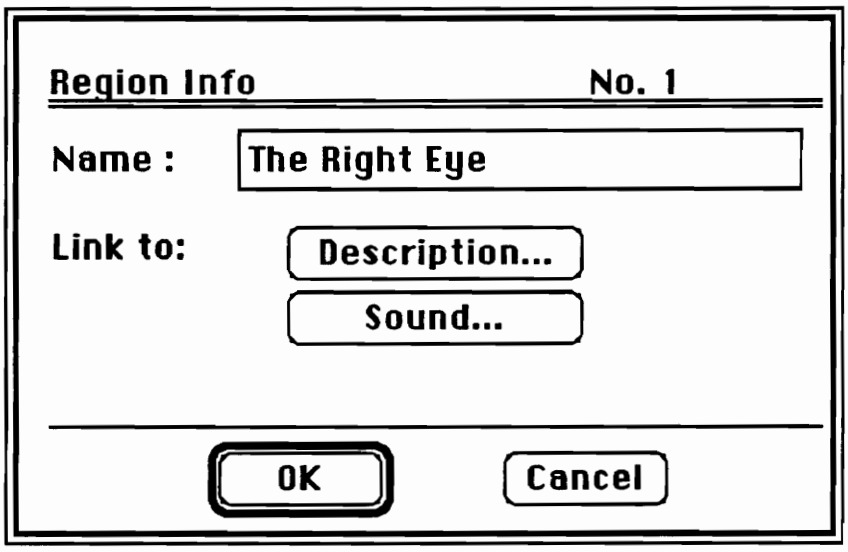


Figure 1.6 The Region Info Dialog

Click in the **Description...** button so we can enter a full-length description for this region. Type in the description as shown in Figure 1.7 and click on the **OK** button.

Now, click on the **Sound...** button of the **Region Info** dialog so we can associate a sound with this region. You will be presented with the Sounds dialog as shown in Figure 1.8. Click once on **TheEye** and select **OK**. This will link the *eye sound* with the *eye region.* See Fig. 1.8.

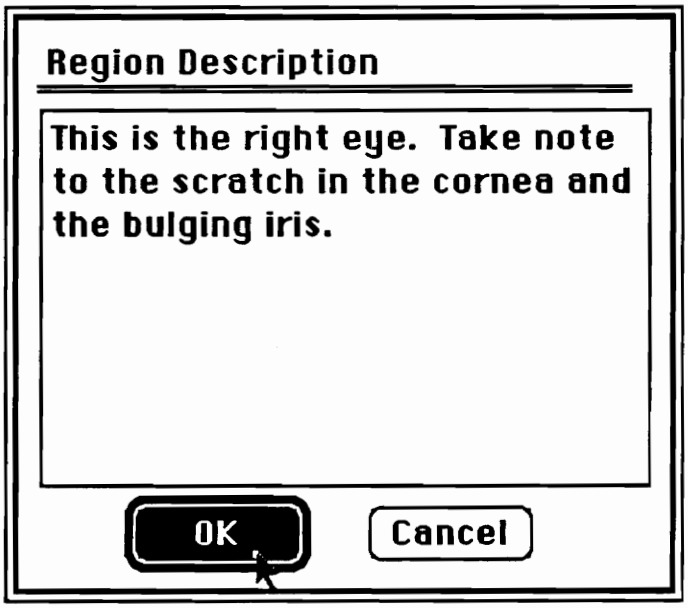


Figure 1.7 The Region Description Dialog

Note: You can hear what the sound sounds like by clicking on it once and selecting the **Play It** button. You can hear as many as you like before clicking OK.

Figure 1.8 The Sound Selection Dialog

Creating Regions with the Box Tool

The Region Info dialog will re-appear. Click OK to close the dialog and to affirm that the label, description and sound are OK. Now, create a second region. This time, we'll use the box tool to create a region around the bird. Select the box tool by clicking once on it. When you move the mouse over the image, the cursor will turn to the cross-hair cursor, + . Move the cross-hair cursor to the upper-left corner of the imaginary rectangle that encloses the bird and hold down the mouse button. As you hold down the mouse button, *drag* the mouse (i.e., move it while holding down the mouse button) to the bottom-right corner of the imaginary rectangle and release the mouse

button. You have just created your second region (see Figure 1.9).

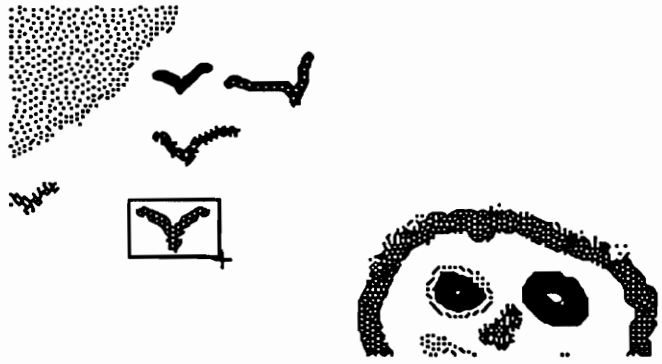


Figure 1.9 Using the box tool to create a region around the bird

As with the eye region, you will be presented with the **Region Info** dialog. For this region, don't enter any label or description, so that we may demonstrate that some regions may only have a sound associated with them. Select the **Sound...** button. Select the **TheBird** sound. When you return to the **Region Info** dialog, click **OK** to accept the information.

Creating Regions with the Polygon Tool

Now, we'll create one more region. We'll use the polygon tool to create a region around the right leg. Select the box tool by clicking once on it. When you move the mouse over the image, the cursor will turn to the crosshair cursor, + . The polygon tool is similar to the lasso tool in that they are both used to create arbitrarily shaped regions. The region created by the lasso is done so by holding the

mouse button down as you outline the region. A region created using the polygon tool is made by clicking on points around the perimeter of the region. Refer to Figure 1.10 below. Place the crosshair cursor on point A. Click the mouse button once and release it. Move the cursor to point B. Click and release. Repeat on each successive point up to point Q. Now, move the cursor over point A (where we started). *Double-click* the mouse button. This *closes* the region.

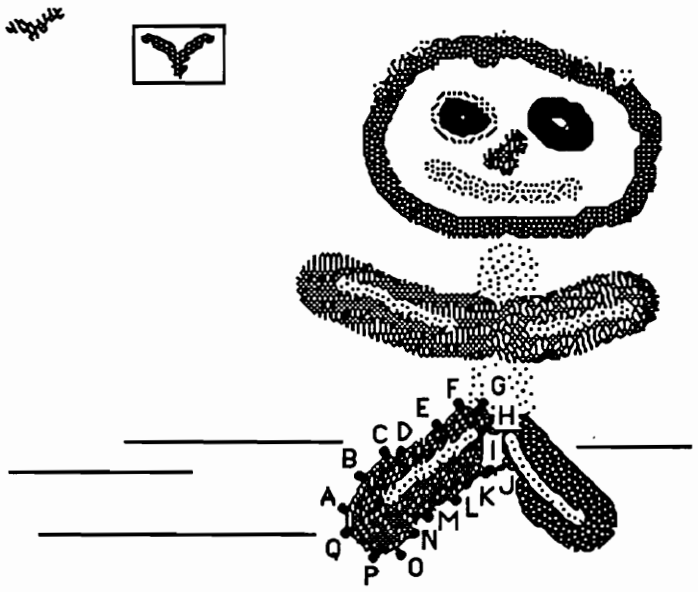


Figure 1.10 Creating a region using the polygon tool

As with the previous regions, you will be presented with the **Region Info** dialog. For this region, you will only enter a name for this region, so that we may demonstrate that some regions may only have a name associated with them. Type **The Right Leg**. Now click in the **OK**

button. You have just created your third region (see Figure 1.11).

> **Note**: You probably have noticed that as you create a new region, it becomes *highlighted* (bordered by a solid black perimeter) and the previously highlighted region becomes *dimmed* (bordered by a gray perimeter). The highlighted region is known as the *current region*. This idea of highlighting is important because the intersect tool, lasso tool, Get Info command and Delete command each work on the current region. You can change the current region by using the arrow tool.

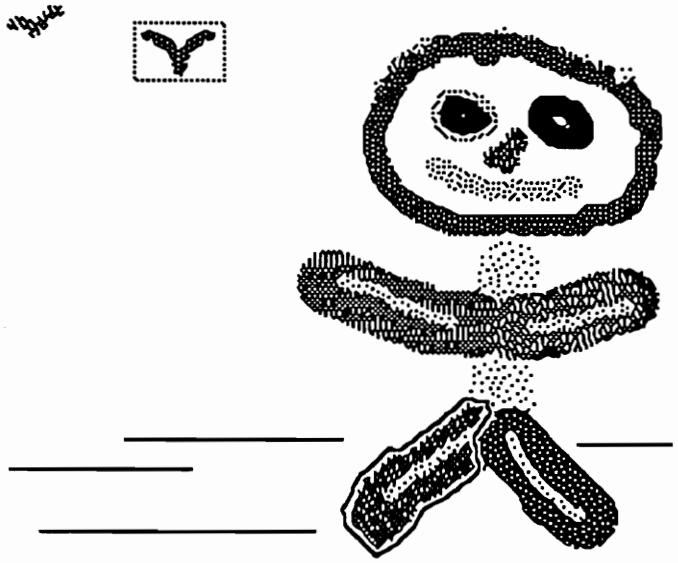


Figure 1.11 Using the polygon tool to create a region around the right leg

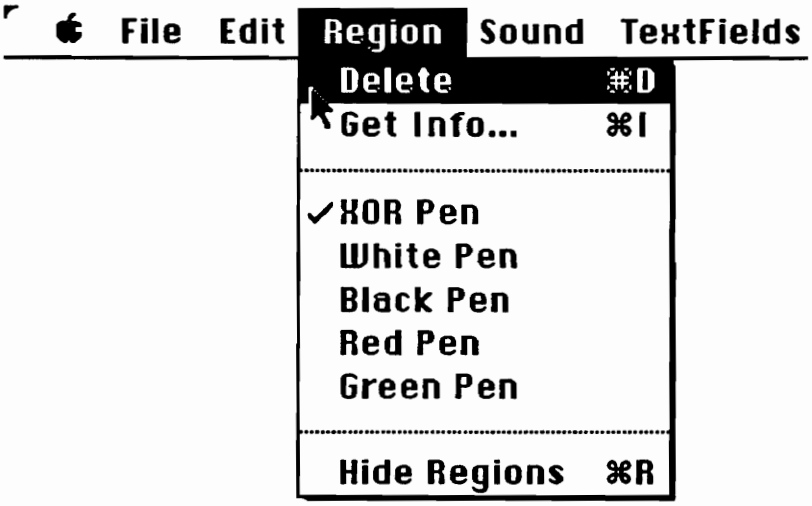
Deleting a Region

Let's create a fourth region so that we can demonstrate the use of the **Delete** command. Select the lasso tool from the **Tools** palette. Move back to the image area and create any region that you like. When the **Region Info** dialog box appears, click in the **Cancel** button (indicating

that we don't want to enter any information for this region). The delete command is used to delete a region. It will delete the currently selected region. Since we want to delete this region that we just created, it should already be the current region (i.e., it should have a solid black perimeter). If it is not, use the arrow tool and click on it.

Select **Delete** from the **Region** menu. The region will be deleted. So, to delete a region:

1. **Make sure it is the current region by clicking on it once with the arrow tool.**
2. **Select Delete from the Region menu.**



Working with Sound and Descriptions

We will now play the associated sounds, lay out the *description boxes* and place links to these boxes. Select the Test Me tool

from the Tools palette. When you move back to the image area, the cursor will change to the question mark, ?. Move the ? cursor (actually the dot of the question mark cursor) anywhere inside the eye region and click the mouse button once. The eye sound will play and the description box will pop up on the screen. See Figure 1.12.

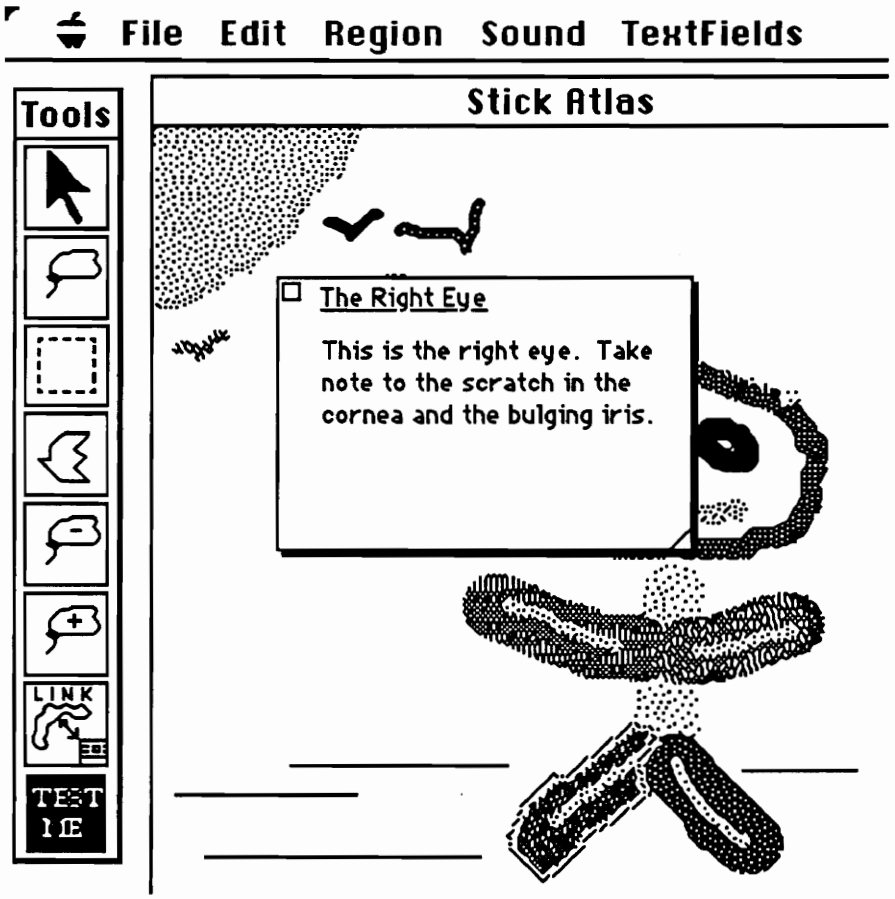


Figure 1.12 Using the Test Me tool to bring up a description box

The Test Me tool is used to place the description boxes appropriately on the screen. When you click in a region with the Test Me cursor for the first time, the Interactive Image Author will place the description box at an arbitrary location. You will have the options of resizing and moving these description boxes. Let's move and resize this box now. Click on the arrow tool in the Tools palette. Move the tip of the arrow into the *resize area* of the description box. See Figure 1.13.

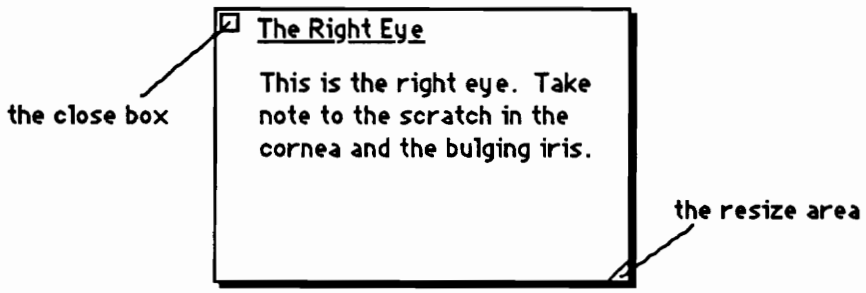


Figure 1.13 The Close Box and Resize Area of a Description Box

When you move the tip into the resize area, hold the mouse button down and *drag* while you hold down the mouse button. As you drag you will notice the outline of the description box change. Release the mouse button when the bottom border lies directly under the last line in the description box. See Figure 1.14.

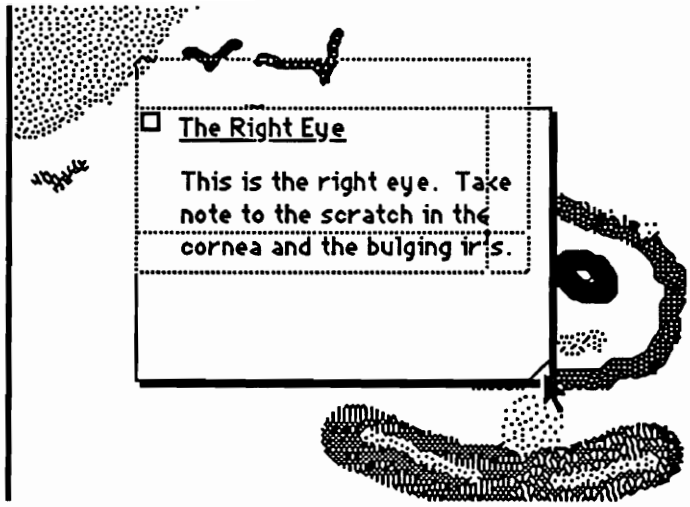


Figure 1.14 Resizing the description box

When you release the mouse button, the description box will be updated to reflect the change in size. See Figure 1.15.

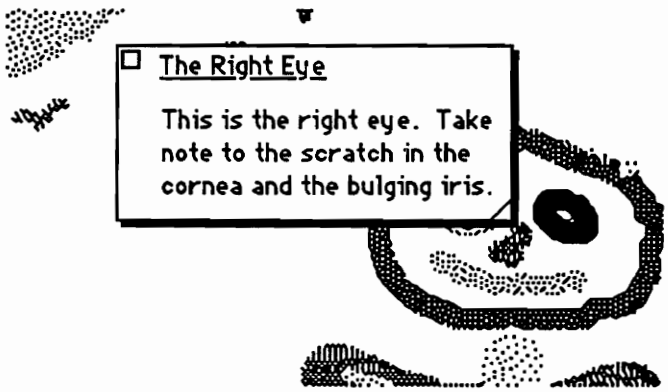


Figure 1.15 A resized description box

Now, let's move the description since it's covering the region that it's describing. To move this description box, position the cursor anywhere in the box *except* in the resize area or the close box. As you

hold down the mouse, *drag* the description box below and partially to the right of the face. As you move, you'll notice the outline of the box follow you. Release the mouse button when the box is in the position shown below. See Figure 1.16.

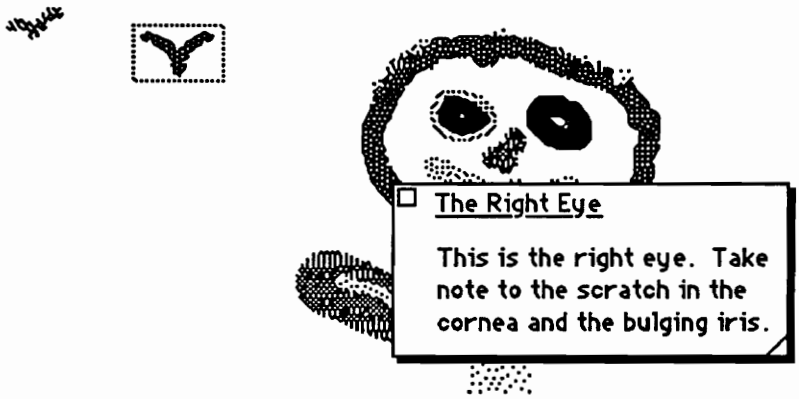


Figure 1.16 The effect of moving a description box

Now, we can lay a link which will link the eye region to its corresponding description box. Click on the link tool

in the Tools palette. The cursor will change to the link cursor . Move the link cursor inside of the eye region. Now drag the mouse until you're somewhere inside its description box. Release the mouse. You have just laid your first link. See Figure 1.17.

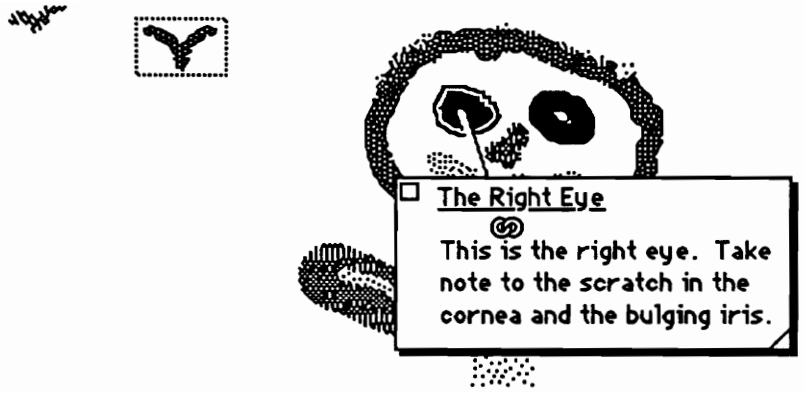


Figure 1.17 Linking a region to its description box

Now select the Test Me tool again. Click inside of the bird region. What happened? Did a description box pop up? Of course not. If you recall, we didn't enter any label or description for this region. However, we did associate a sound for this region so you should have heard a voice stating that this is, in fact, the bird. You can click in the region as often as you like.

Click in the leg region. Did you hear anything? You shouldn't have. We only associated a label for this region. Resize and move the description box as you did above. Use the link tool to place a link between the leg and description box. You should then have an interactive image that looks something like Figure 1.18.

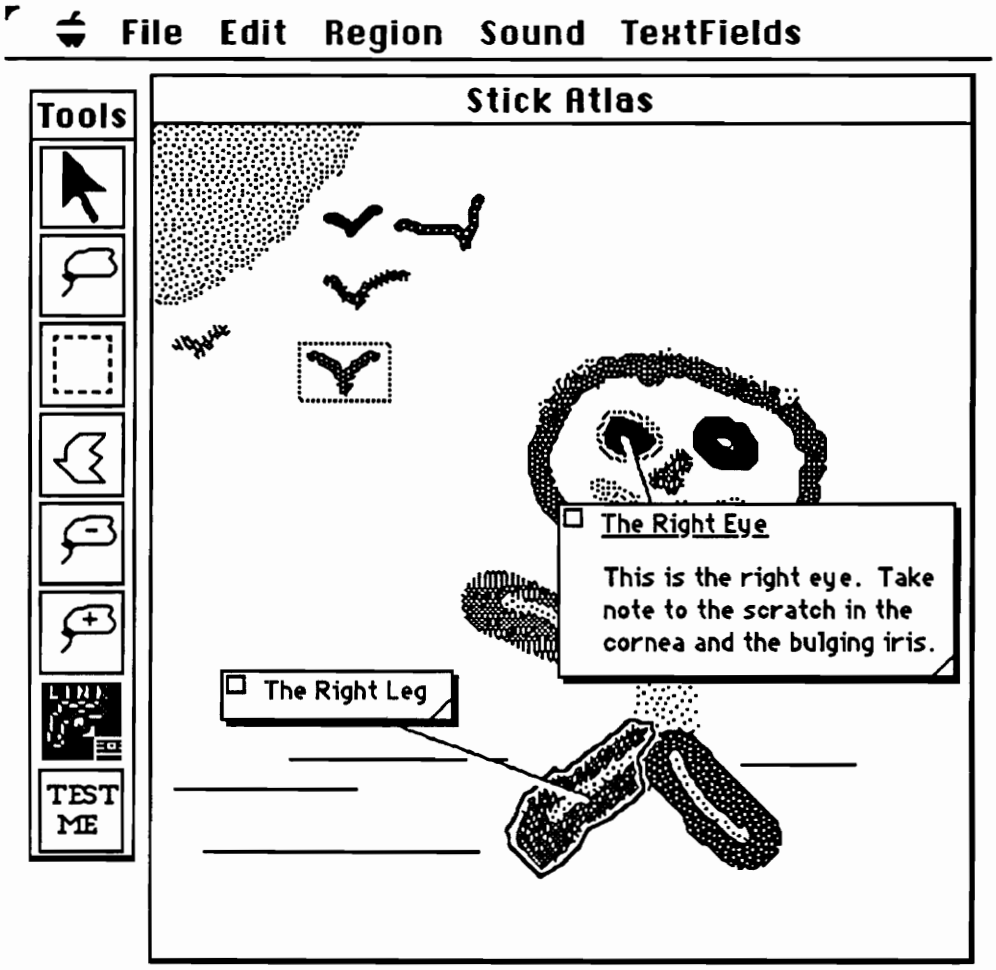


Figure 1.18 An image with three regions and two description boxes

Note: If the region has a label and description associated with it, the label will be underlined in its description box. If it only has a label, the label will NOT be underlined. Refer to the two description boxes on the screen.

Saving your Image

Let's save your first interactive image creation. Select **Save** from the **File** menu. The Save dialog box will appear. Give this file a name, such as Stick Man Atlas and click on the **OK** button. See Figure 1.19.

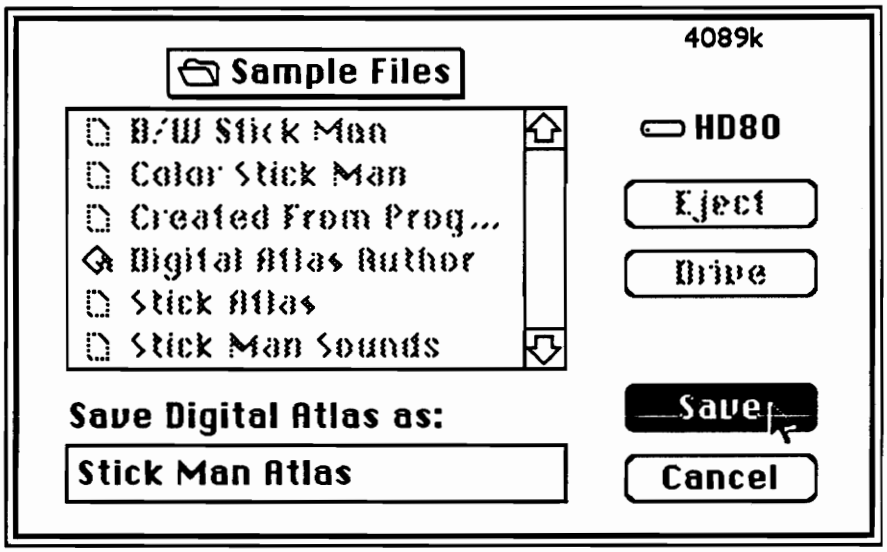
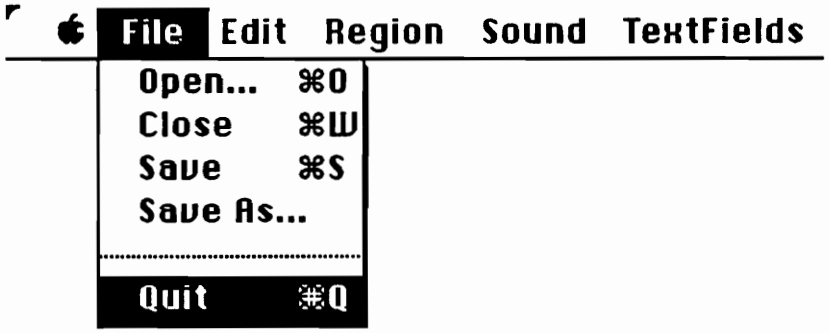


Figure 1.19 Saving an interactive image file

Congratulations! You have successfully created your first interactive image. To quit Interactive Image Author, select **Quit** from the **File** menu as shown below.



Chapter 2

The Interactive Image Author

What is an Interactive Image?

The Interactive Image Author program is an authoring tool that allows the user to open a graphic file and add content to it. This content consists of key regions that the user creates by using MacPaint™-like tools. Optional descriptions and sounds are then associated with these regions. The graphic file with the content added to it is called an *Interactive Image*, simply referred to as image from here on. Image files are saved to disk and are utilized by the Quiz program. The image concept can be applied to many fields of interest. For example, a geography quiz lesson can be developed by obtaining digital maps of the world. Using the Interactive Image Author program, each state can be regioned off and information about each state can be entered. The quiz program can then ask questions such as "Please click on the state whose capital is Bismarck." The Interactive Image Author program and the quiz programs were designed with user interaction

in mind. As another example, consider the medical field. The Interactive Image Author can be used to open a digital image of the brain. The pituitary gland, cerebrum, cerebellum, etc... can be regioned off and the file saved as an interactive image. A quiz can be constructed to ask medical students questions to identify given regions or ask them to find particular regions of interest.

Every interactive image file must begin from a graphic file. The process of creating an interactive image file is:

- A graphic file is loaded.
- Regions of interest are created using the new region tools.
- The regions are touched up using the edit region tools.
- Optional labels and descriptions are entered for each region.
- Optional sounds are associated for each region.
- Optional links are drawn between the region to the description.

The Use of Monochrome, Gray-scale and Color

The Interactive Image Author can be used on any Macintosh. That includes every Macintosh from the Macintosh 128 all the way up to the latest and most powerful Macintosh to date, the Macintosh IIfx. This means that any monochrome, gray-scale or color PICT file types can be opened with the Interactive Image Author program. Currently,

Interactive Image Author only supports the PICT file type. PICT is a standard Macintosh type. Virtually all Macintosh graphic programs support the PICT type, such as Studio/8, PixelPaint, Canvas and SuperPaint, among others.

This means you can create any color, gray-scale or monochrome image and save it with the PICT type. Interactive Image Author can open it and display it using the current monitor settings of your **Control Panel**. You can even open a color PICT file on a monochrome screen. A conversion takes place, and the image will be displayed in two colors, black and white. If you open a color file on a gray-scale screen, the colors are converted to appropriate levels of gray, and the image is displayed in gray-scale.

Chapter 3

Regions

Creating Regions

An interactive image file is constructed by taking a graphic file and creating regions of interest. The Interactive Image Author enables you to create regions by using any one of the three region tools. They are the **lasso**, the **box** and the **polygon**. Using these tools, you can create regions of any shape and size.

You can use the lasso to create any type of region. Refer to Figure 3.1 which shows three regions that were created by the lasso tool.

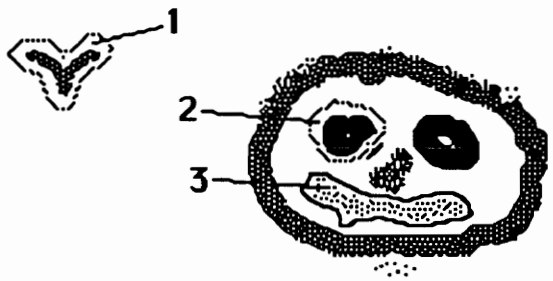


Figure 3.1 Three regions created with the lasso tool

The box and polygon tools are available to make it easier to create certain types of regions that could be created by the lasso tool but

would be more difficult to do. It is tedious to create a rectangular region using the lasso. You would need a steady hand to keep straight on the edges. Using the box tool, it's as easy as clicking on one corner of the box and dragging to the opposite corner of the box. Refer to Figure 3.2 which shows four regions created by the box tool.

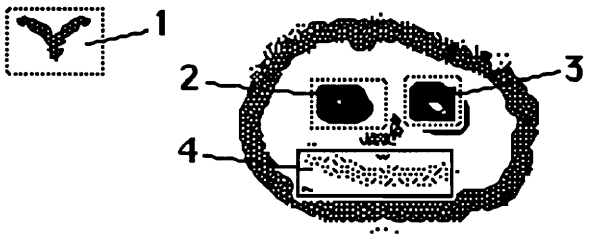


Figure 3.2 Four regions created with the box tool

The polygon tool is useful for creating regions that are dominated by straight edges along its perimeter such as a star. This does not mean that the polygon tool can only be used to create straight edge regions. The polygon tool can be used to create any region that the lasso can create. In fact, the polygon tool can be approximated to be as functional as the lasso tool by keeping the line segments to minimal length. You may find yourself using only the polygon tool to create all of your regions. It is a matter of choice for the majority of the regions you will need to create. There are regions, however, where it is clear that the polygon tool should be used to create the region. Refer to Figure 3.3 which shows three regions created by the polygon tool.

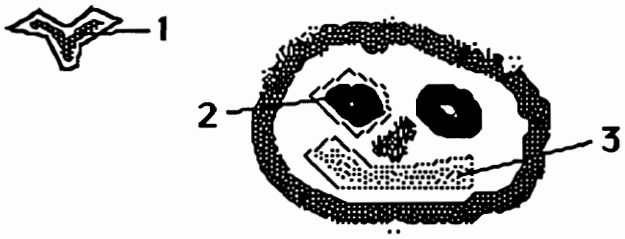


Figure 3.3 Three regions created with the polygon tool

We will now go through each tool in detail discussing the mechanics of creating each type of region.

The Lasso Tool

The lasso tool is used to create free-form regions. To create a region with the lasso tool, first click on the lasso tool located in the tools palette. Choose any point on the perimeter of the region that you are about to create. Move the mouse to position the cursor over the image area. Notice how the cursor changes from the arrow to the lasso . Position the tip of the lasso on the point that you chose on the perimeter. Click and hold down on the mouse button. As you hold the button down, outline the region. When you return to the starting point, let go of the mouse button.

Note: You don't have to return EXACTLY on your starting point. You can release the button at any point before. The region will close up by drawing a straight line from where you released the button to the starting point.

If the region didn't turn out the way you wanted, don't worry. You can edit the region using edit region tools as described later in this chapter. When you release the mouse button, you will be presented with the Region Info dialog.

While creating a region, it is possible to release the mouse button too soon. If this occurs, there are two things you can do. You can immediately select **Delete** from the **Region** menu to delete the region, then create the region again; or, use the edit region tools to edit the incomplete region as described in the Editing Regions section later in this chapter.

The Box Tool

The box tool is used to create rectangular-only regions. To create a region with the box tool, first click on the box tool located in the tools palette. The region is creating by *clicking* on any one of the four corners of the rectangular area and *dragging* to the opposite corner. Choose any corner of the rectangular region that you are about to create. Move the mouse to position the cursor over the image area.

Notice how the cursor changes from the arrow to the cross-hair cursor + . Position the center of the cross-hair cursor on the corner that you chose. Click and hold down on the mouse button. As you hold the button down, move the cursor to the opposite corner (this is called *dragging*). Now, release the mouse button. If the region didn't turn out the way you wanted, don't worry. You can edit the region using edit region tools as described later in this chapter. When you release the mouse button, you will be presented with the Region Info dialog so you may enter a label for the region and descriptive information.

While creating a region, it is possible to release the mouse button too soon. If this occurs, there are two things you can do. You can immediately select **Delete** from the **Region** menu to delete the region, then create the region again; or, use the edit region tools to edit the incomplete region as described in the Editing Regions section later in this chapter.

The Polygon Tool

The polygon tool is used to create irregularly shape regions. Although the lasso tool can do the same, the polygon tool is helpful for regions that have straight-edge sides. A region is created with the lasso by moving the cursor to any point on the perimeter of the region,

clicking on the mouse button, outlining the region (as you hold down the mouse button), and releasing the mouse button when returning to the original point. A region is created with the polygon tool by repeated steps of pointing and clicking around the perimeter of the region.

To create a polygon region, first click on the polygon tool located in the tools palette. Choose any point on the perimeter of the region that you are about to create. Move the mouse to position the cursor over the image area. Notice how the cursor changes from the arrow to the cross-hair cursor + . Position the center of the cross-hair on the point that you chose on the perimeter. Click on the mouse button. This sets the first vertex of the polygon region. As the cursor is moved away from the first vertex, a line segment is drawn from the first vertex to the current mouse position. Choose a point near the first that will be the second vertex of the region. Move the mouse to that position and click the mouse button. Repeat this procedure of pointing and clicking until you return to the first vertex that you created. When the cursor is over the first vertex, *double-click* the mouse button by clicking twice in rapid succession without moving the mouse. If the region didn't turn out the way you wanted, don't worry. You can edit the region using edit region tools as described later in this chapter. When you release the mouse button, you will be

presented with the **Region Info** dialog.

While creating a polygon region, it is possible to double-click the mouse button by accident before reaching the first vertex. You can immediately select **Delete** from the **Region** menu to delete the region, then create the region again; or, use the edit region tools to edit the incomplete region as described in the Editing Regions section which follows this section.

Editing Regions

Very often, you will find that the region didn't turn out exactly as you had hoped. As you probably have noticed, it's easy to move the mouse further than you want. If this happens to you, use the edit region tools to edit the region. There are two edit region tools, the **Union Region Tool** and the **Intersect Region Tool**. The union region tool is used to append to a region. The intersect tool is used to delete a piece of a region.

For example, suppose you were trying to use the lasso tool to create a region around the state of Florida as shown in Figure 3.4a and you created the region as shown in Figure 3.4b.

Figure 3.4 An improper region around Florida created by the lasso tool

As you can see, the region extends too far into the Atlantic Ocean and the region doesn't completely enclose the panhandle of Florida. We'll use the intersect tool to delete the piece of the region that extends into the Atlantic. We'll use the union tool to append the area of the panhandle that was left out of the region.

Select the intersect tool from the tools palette. Move the cursor over the image area. The cursor will change from the arrow to the intersect cursor, . It resembles the lasso but it has a minus sign inside of the lasso loop. Move the tip of the intersect cursor to any point *outside* of the region but near the area that will be deleted. See Figure 3.5a. As you hold down the mouse button, enclose the area that will be deleted (remember to keep the mouse button down). See Figure 3.5b. Return to the starting point. Release the mouse button. An intersection will be performed between the region of interest and the area you just enclosed. The resulting region from the intersection will remain. See Figure 3.5c.

Note: You probably have noticed that as you create a new region, it becomes *highlighted* (bordered by a solid black perimeter) and the previously highlighted region becomes *dimmed* (bordered by a gray perimeter). The highlighted region is known as the *current region*. This idea of highlighting is important because the Intersect Tool, Lasso Tool, Get Info command and Delete command each work on the current region. You can change the current region by selecting the arrow tool and clicking on the region that you would like to be the current region.

Figure 3.5 Using the intersect tool to delete a piece of a region

Therefore, to intersect, or *chop off* a piece of a region:

1. Make sure that the region is the current region by selecting the arrow tool and clicking once on the region. It should become highlighted (i.e., a black border should surround the perimeter of the region; non-current regions have a gray border around them).

2. Select the intersect tool from the tools palette.

3. Outline a new region that will be intersected with the region of interest so that the intersection will cause the area of concern to be deleted. When you release the mouse button, the intersection will be

performed and the resultant region will remain.

To append a region, select the union tool from the tools palette. Move the cursor over the image area. The cursor will change from the arrow to the union cursor, . It resembles the lasso but it has a plus sign inside of the lasso loop. Move the tip of the union cursor to any point *inside* the region but near the area that will be appended. For example, see Figure 3.6a. As you hold down the mouse button, enclose the area that will be appended (remember to keep down the mouse button). See Figure 3.6b. Return to the starting point. Release the mouse button. A union will be performed between the region of interest and the area you just enclosed. The resulting region from the union will remain. See Figure 3.6c.

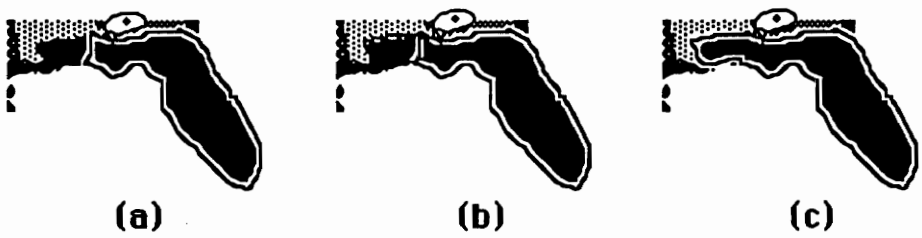


Figure 3.6 Using the union tool to append an area to a region

Therefore, to *append* an area to a region:

1. Make sure that the region is the current region by selecting the

arrow tool and clicking once on the region. It should become highlighted (i.e., a black border should surround the perimeter of the region; non-current regions have a gray border around them).

2. Select the union tool from the tools palette.

3. Outline a new region that will be unioned with the region of interest so that the union will cause the newly outlined area to be appended to the current region. When you release the mouse button, the union will be performed and the resultant region will remain.

Deleting Regions

Very often, you will find that the region didn't turn out exactly as you had planned. If this happens to you, there are two things you can do. You can use the edit region tools and try to *repair* the region to what you want it to be, or you can simply delete the region and create it again. They will both do the job. It is a matter of choice.

The **Delete** command from the **Region** menu is used to delete a region. It will delete the currently selected region. Make sure that the region you want to delete is the current region. If it is not, use the arrow tool and click on it. Select **Delete** from the **Region** menu. The region will be deleted.

Note: Deleting a region also deletes its associated link, label and description information.

So, to delete a region:

1. Make sure it is the current region by clicking on it once with the arrow tool.

2. Select **Delete** from the **Region** menu. (see Figure 3.7)

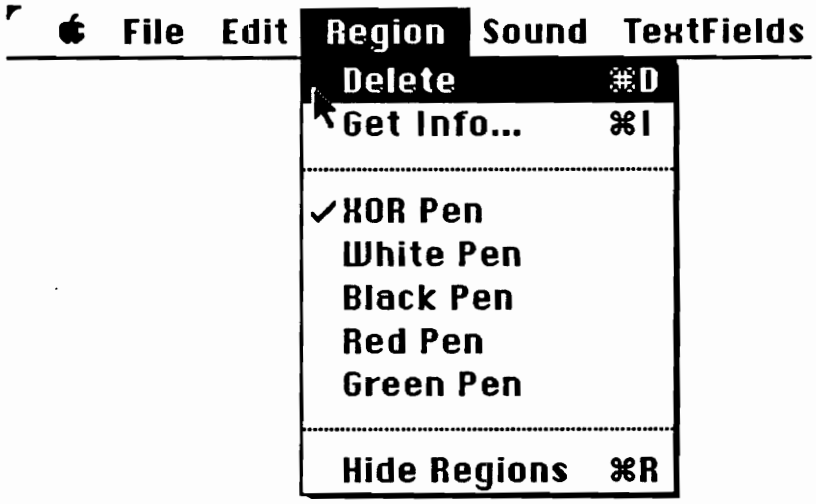


Figure 3.7 Selecting the Delete command from the Region menu

Moving Regions

To move a region, select the arrow tool from the tools palette. Click on the region that you would like to move. As you hold down the

mouse button, *drag* the region to its destination and release the mouse button.

The Use of Pen Color

When you create regions, you have the option of using one of five pens. Most of the time you will want to use the XOR pen. XOR is an acronym for *exclusive-or*. As you draw (lasso regions, box regions, polygon regions, union regions, or intersect regions) with an XOR pen, each pixel drawn will be the inverse of the pixel that used to be there. For example, suppose you were working with an image that had a white lake bordered by black land. See Figure 3.8.

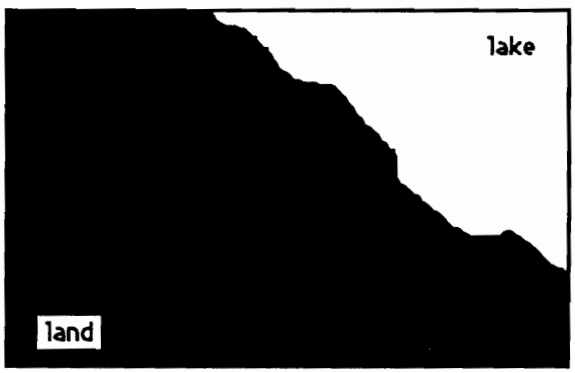


Figure 3.8 Sample image to demonstrate the use of the XOR pen

If you drew a region that included a portion of the lake and a portion of land, the perimeter of the region that was on the land would be drawn in white and the perimeter that was on the lake would be drawn in black. See Figure 3.9.

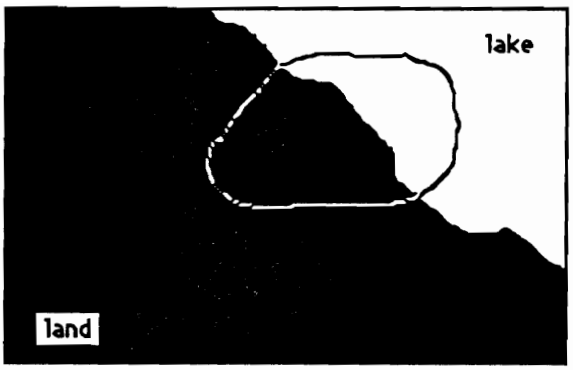


Figure 3.9 A region created using the XOR pen

The XOR pen works well on a monochrome or gray-scale screen. However when color is used, the inverse of some colors is hard to distinguish and there is difficulty in creating the region. If this is the case, you can choose a solid color pen. The colors available are **White**, **Black**, **Green**, and **Red**. They are available under the **Region** menu. See Figure 3.10.

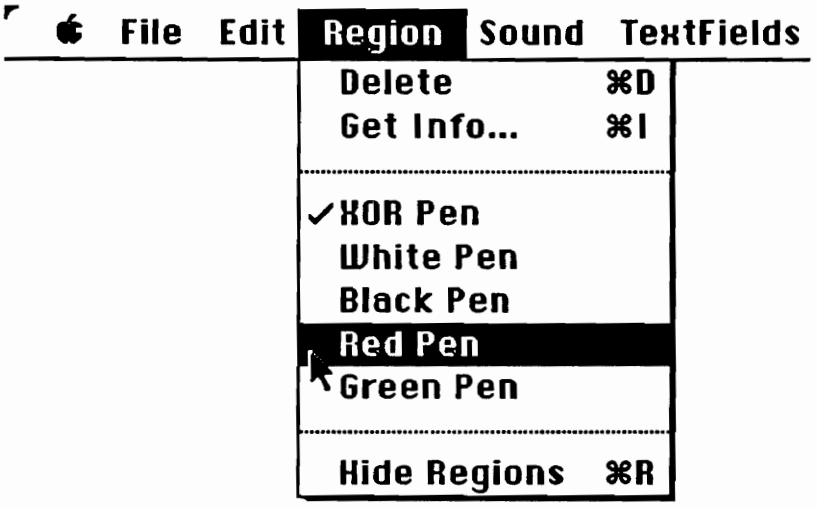


Figure 3.10 Selecting a different pen color under the region menu

Certain image conditions warrant the use of one pen over another. However, you may find that you always use the XOR pen. The other pen colors are there if needed.

> **Note**: The choice of pen color is available for aesthetic reasons only. The color information is not saved along with the regions. The pen color is only used when creating a region or editing a region.

Chapter 4

Labels and Descriptions

Creating Regions

Whenever a region is created, the **Region Info** dialog is presented to the user. See Figure 4.1. This dialog enables the user to enter an *optional* label and description for the region just created.

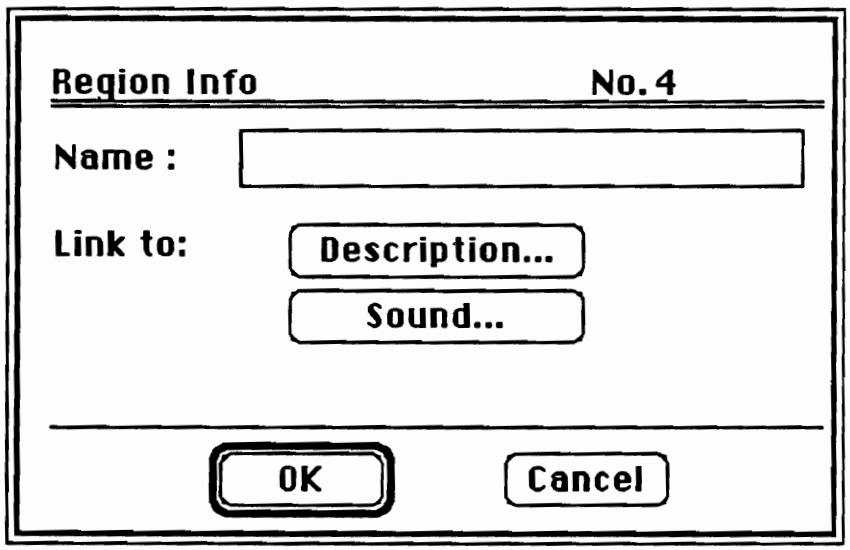


Figure 4.1 The Region Info Dialog

The label and description are optional. This information is displayed to the user in the quiz module when they click on the region. When

the user clicks on a region with a label and description associated with it in the quiz program, a description box will *pop up* displaying the label and description. See Figure 4.2 which shows part of an interactive image file that has two description boxes that were *popped up*.

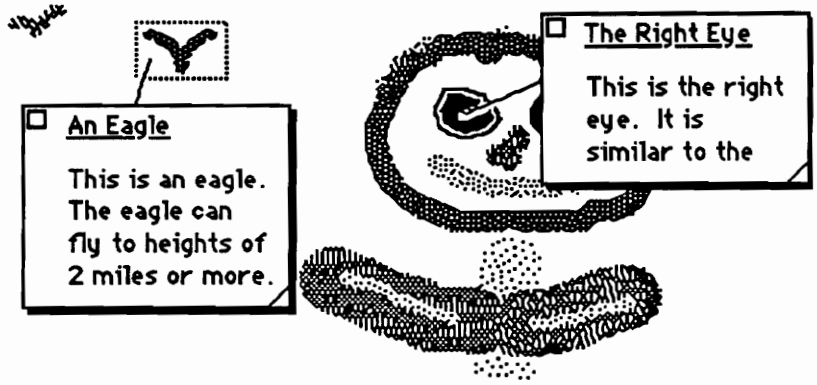


Figure 4.2 A typical image with open popped-up description boxes

As stated above, the **Region Info** dialog will be presented to the user after creating a region. To assign a label to a region, type the label in the **Name** field of the dialog. To assign a description, click in the **Description...** button. The **Region Description** dialog will appear as shown in Figure 4.3.

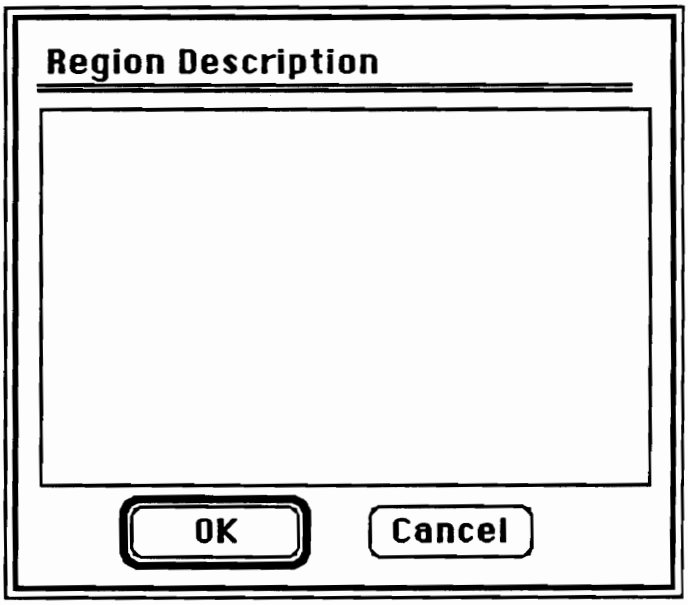


Figure 4.3 The Region Description Dialog

Type in the description and click on the **OK** button. This will associate the description to the region.

Note: You can also press the Enter key as a shortcut to clicking in the OK button.

You will then return to the **Region Info** dialog. Click on the **OK** button (or press Return for a shortcut) to accept the label and description for this region.

Modifying Labels and Descriptions

Once the label and description has been associated for a region, it remains so until the region is deleted or the label and description is

modified. To modify the label and description for a region, first make that region the *current region.* That is, select the arrow tool from the Tools palette and click once on the region. A region is the current region if it is bordered by solid black. All other regions will be bordered by gray. The next step is to select **Get Info** from the **Region** menu (or press ⌘-I as a shortcut). The **Region Info** dialog will appear as shown in Figure 4.1. The current label will be in the name field. Use the standard Macintosh editing techniques to edit the label. To edit the description, click in the **Description** button and the **Region Description** dialog will appear as in Figure 4.3. The current description for this region will appear in the field. Use the standard Macintosh editing techniques to edit the description. Click in the **OK** button to accept your changes. If you want to get back what you had previously for the description, click **Cancel** instead. When you return to the **Region Info** dialog, click in the **OK** button to accept your changes for this region.

Description Boxes

A description box is used to display the label and description associated for a given region. The Test Me tool

from the Tools palette is used to display the description boxes. When you select

the Test Me tool and move back to the image area, the cursor will change to the question mark, ?. If you move the ? cursor anywhere inside a region that has a label, description, or sound associated with it and click the mouse button once, the associated sound will play and the description box will pop up on the screen.

The Test Me tool is used to place the description boxes appropriately on the screen. When you click in a region with the Test Me cursor for the first time, the Interactive Image Author will place the description box at an arbitrary location. You will have the options of resizing and moving these description boxes. To move and resize the description box, click on the arrow tool in the Tools palette. Move the tip of the arrow into the *resize area* of the description box. See Figure 4.4.

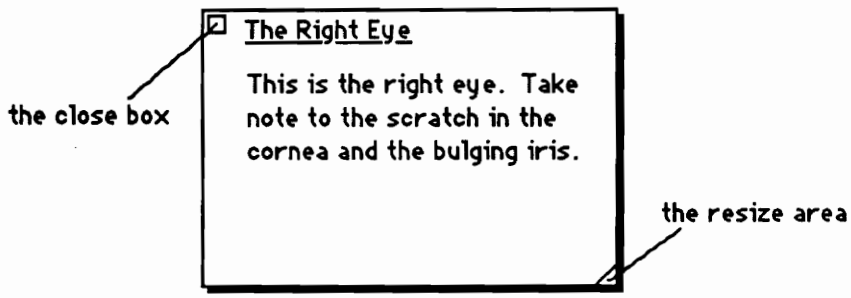


Figure 4.4 The Close Box and Resize Area of a Description Box

When you move the tip into the resize area, hold the mouse button down and *drag* while you hold the mouse button down. As you drag you will notice the outline of the description box change. Release the mouse button when the outline of the description box is the size that

you like. See Figure 4.5.

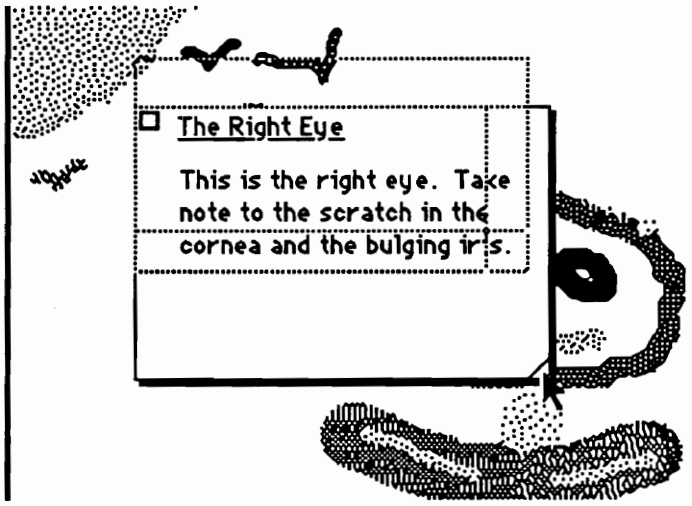


Figure 4.5 Resizing the description box

When you release the mouse button, the description box will be updated to reflect the change in size. See Figure 4.6.

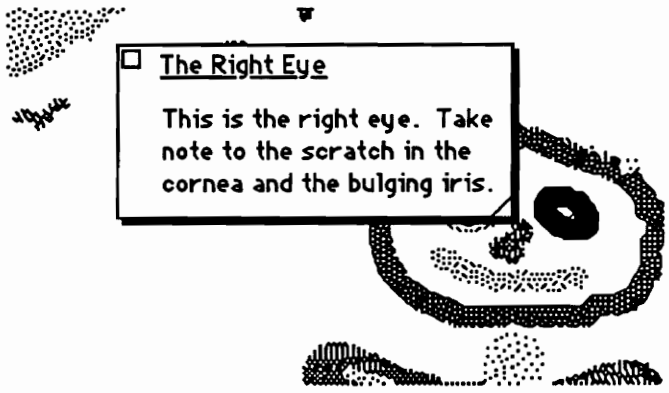


Figure 4.6 A resized description box

If the description box covers the region that it's describing, you can easily move the description box. Position the cursor anywhere in the box *except* in the resize area or the close box. As you hold down the mouse, *drag* the description box to its new position. As you move, you'll notice the outline of the box *follow* you. Release the mouse button when the outline of the box is where you would like it to be. See Figure 4.7.

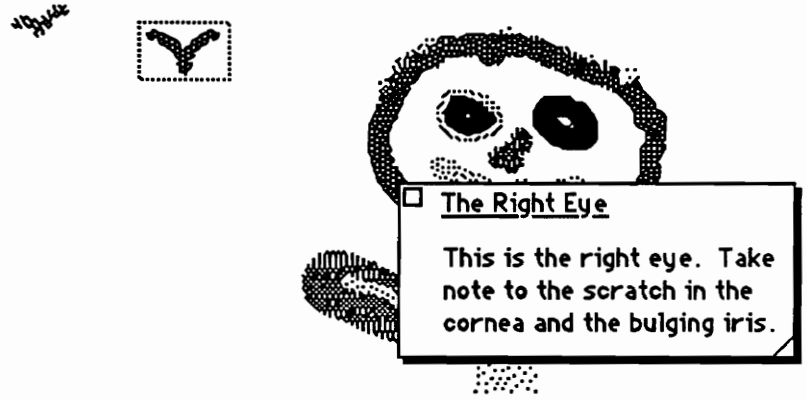


Figure 4.7 The effect of moving a description box

Chapter 5

Links

Creating a Link

Links are used to provide a *visual link* between a region and its corresponding description box. They are optional, and you may find that some description boxes are appropriate without a link associated with it.

Click on the link tool

in the Tools palette. The cursor will change to the link cursor .

To associate a link with a description box, make sure that the description box is *popped up* on the screen. If it is not, use the Test Me tool to display it (see Chapter 4 on description boxes). Move the link cursor inside of the region of interest. Now drag the mouse until you're somewhere inside its description box. Release the mouse. The link is now associated with the region. See Figure 5.1.

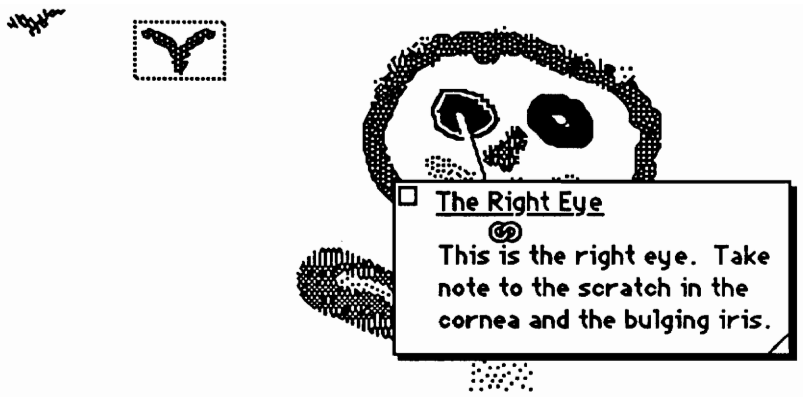


Figure 5.1 Linking a region to its description box

Note: If you attempt to draw a link but no link is drawn after releasing the mouse button, this indicates that you are trying to link a region with another regions description box. Make sure that you draw the link from the region to its own description box.

Modifying a Link

To modify a link, you simply create a new one over the old one. Move the link cursor inside of the region of interest. Now drag the mouse until you're somewhere inside its description box. Release the mouse. The new link is now associated with the region. The screen is updated to reflect the fact that the old link has been removed and the new link is drawn.

Deleting a Link

To delete a link, first make the region, whose link you want to delete, the current region (select the Arrow tool from the Tools Palette and

click once on the region). After selecting the region as the current region, select **Delete Link** from the **Textfields** menu. The link is removed.

Chapter 6

Working with Sound

Creating Sounds

The Interactive Image Author works with any sounds of resource type 'snd .' This is the standard Macintosh sound resource format. Farallon Computing markets a device called the MacRecorder™ (Farallon Computing, $159) which allows you to *digitize* your own voice as well as any sound effects. It is sold with software that allows you to create sound files that will work with the Interactive Image Author. This manual assumes that you have access to MacRecorder, or any other program that will save the sounds in the 'snd ' format.

The sounds are optional but they add a new dimension to the interactive image file. When you associate a sound to a region, the sound will be played when the user clicks on it. It is a useful feature for images such as a foreign language image. For example, you could design Spanish interactive image screens. There could be some Spanish items on the screen with their English label beneath them. You can associate the Spanish word for each object as the label and

associate the pronunciation of that Spanish word as the sound for that region. So, the user can *browse* around the image by clicking on an object and hearing how the word is pronounced in Spanish.

To associate sounds to your regions on an interactive image file, you need to *select a sound file* for your interactive image.

> **Note**: You can only associate one sound file for each image. If there are sounds that you want to use that are located in multiple files, you need to use a program such as ResEdit™ from Apple Computer that allows you to copy and paste resources from one file into another. In this manner, you can create a new sound file that will contain all of the sounds from your other files.

Selecting the Sound File

Before you can associate a sound to a region, you must select a sound file. (You must have your image open first, before you can do this). Choose **Select Sound File...** from the **File** menu (or press ⌘-F) to associate a sound file with your image.

You will then be presented with an *open dialog box.* It is the standard

Macintosh dialog for selecting a file.

A dialog should appear on your screen as shown in Figure 6.1. Find the sound file which contains the sounds that you will use for this image and **Open** it.

Now that your sound file is associated with your image, you can associate sounds from this sound file to any of the regions that you create.

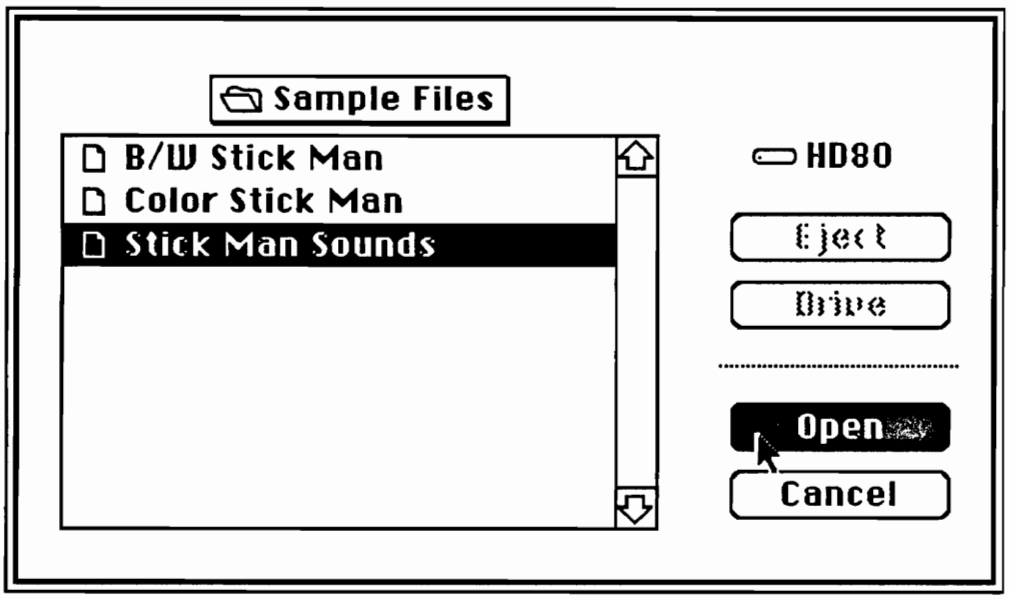


Figure 6.1 A typical Select Sound File dialog

Associating Sound to a Region

After creating a region using one of the lasso, box, or polygon tools,

you will be presented with the Region Info dialog box. This dialog allows you to associate a label to a region, associate a description to the region, and associate sound to a region.

When the Region Info dialog is on the screen, you can associate a sound to a region by clicking in the **Sound...** button. You will be presented with the sounds dialog similar to the one shown in Figure 6.2. Click once on the sound of your choice and click on the **OK** button. This will associate your sound with the current region.

Note: You can hear what the sound sounds like by clicking on it once and selecting the Play It button. You can hear as many as you like before clicking OK.

Figure 6.2 A typical Sound Selection dialog

To change the sound associated with a region to another sound, make the region the current region by clicking on it once with the arrow tool. Select **Get Info...** from the **Region** menu. The **Region Info** dialog will appear. Click on the **Sound...** button. The **Select Sound** dialog will appear. The sound currently associated with this region will be highlighted. Select a new sound and click on the **OK** button. The **Region Info** dialog will again appear. Click on the **OK** button to accept your change.

> **Note**: Instead of clicking on the region and selecting Get Info... from the Region menu, you can simply double-click on the region as a shortcut.

Chapter 7

File Management

Saving an Interactive Image

To save your interactive image file, select **Save** from the **File** menu. The Save dialog box will appear. Type a name for this file obeying Macintosh conventions for file names and click on the **OK** button. See Figure 7.1.

> **Note**: Although the image does contain a PICT image in it, it cannot be read by any of the popular Macintosh paint programs such as Studio/8, PixelPaint, MacPaint, MacDraw, Canvas, Image, SuperPaint, etc. Refer to Appendix B for the interactive image file format specification.

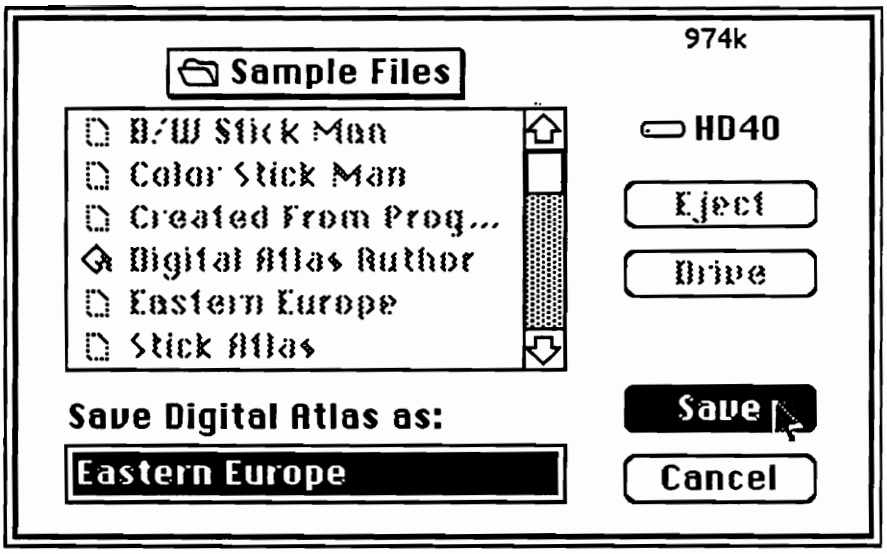


Figure 7.1 Saving an interactive image file

The file will then be saved to disk as an image file and can be read by the Interactive Image Author at a later date or it can be read by the Quiz program or any other program that will read interactive image formatted files.

> **Note**: Like any thoughtful Macintosh program, if you try to quit the program without saving all of your files, Interactive Image Author will give you a warning that the image has not been saved and you are given a chance to save it.

Opening an Image

To open an image file, select **Open** from the **File** menu. The Open dialog box will appear as shown in Figure 7.2. *(Note: the exact listing of files shown here will probably be different from the listing shown on your screen).*

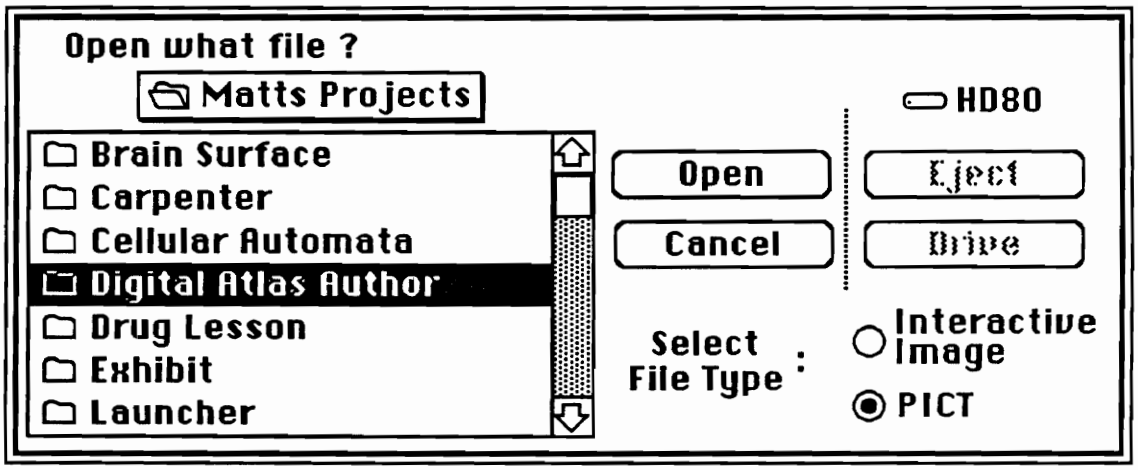


Figure 7.2 The Open Dialog Box

This Open dialog is used for opening both PICT files (which are made into interactive image files after content is added to it) and interactive image files. Click in the *radio button* located to the left of the Interactive Image type. The file list will only display interactive image files and folders (when the PICT radio is selected, only PICT files and folders are displayed).

Locate and click on the interactive image of your choice. Click on the **Open** button.

The interactive image file will be read, a little tune will play and the Tools palette will appear to the left of a window containing the interactive image file.

Working with Multiple Interactive Image Files

The Interactive Image Author has the capability of working with more than one interactive image file. You can work with up to five image files at one time. After opening the first image file, you may select **Open** from the **File** menu to open four other files. The image windows will be *stacked* over one another. Only one image window is considered the *active window* at any time. It is the window whose title bar is filled with horizontal lines. To make an image window the active window, use the arrow tool and click on any visible portion of the window (i.e., the windows title bar or the image area). The current active window will become inactive (the title bar is filled in white) and the window just clicked on becomes the active window (the title bar is filled with horizontal lines and it is brought in front of all other windows).

Appendix B

The Interactive Image File Specifications

Interactive Image File Structure

The interactive images are kept in memory in a structure as shown below:

```
struct windowListRec {
      Boolean     dirty;
      /* Used to determine if the file was changed since last save */

      Str255      sndFileName;
      /* Sound file associated with this image. */

      Str255      theFileName;
      /* Keep track of filename for SFPut... and SFGet... */

      Boolean     used;
      /* When an image is open, it's 'used' value is TRUE.
         When an image is closed, it's record position in this
         structure becomes available, so used=FALSE. */

      WindowPtr   theWPtr;
      /* The pointer to the window structure used by this image */

      Rect  theBounds;
      /* The bounding rectangle of this image */

      PicHandle   thePictHdl;
      /* A handle (ie, ptr to a ptr) to the PICT structure used
         by this image */

      int   ttlRegions;
      /*  No. of regions used by this image. */

      int   currRgnSel;
      /* The current region selected (0 if none). */

      RgnHandle   rgnList[MAX_REGIONS];
      /* The array of regions for this image. */

      Ptr   label[MAX_REGIONS];
      /* Label associated with each region. */

      int   theSnd[MAX_REGIONS];
      /* Sounds associated with each region. */
```

```
      WindowPtr   txtWPtr[MAX_REGIONS];
      /* The array of window pointers for each of the description
         boxes for this image. If a region has no text box, then
      its corresponding ptr is NIL. */

      Boolean     txtWOpen[MAX_REGIONS];
      /* Used to keep track if the description box is currently
         open. */

      Boolean     autoShow[MAX_REGIONS];
      /* Currently not used. Each region has an autoShow bit
         associated with it to determine if the description box
         should automatically pop up when the image is opened
         in the quiz program. */

      Rect  txtWBounds[MAX_REGIONS];
      /* The bounding rectangles for each of the description boxes*/

      Rect  txtWLink[MAX_REGIONS];
      /* The links for each of the description boxes. The upper-left
         corner of the rect is the start pt of the link and the
         lower-left corner of the rect is the end point to the
         link. */

      Ptr   txtWText[MAX_REGIONS];
      /* The text for each of the description boxes. */

} windowList[maxWindows];
```

The structure of the interactive image data file is shown below:

int	the size of the picture
binary	the picture data
Str255	the sound file name associated with this image
int	total regions
Rect	the bounding rectangle for this image window

<for each region:>

Str255	the label for this region
int	the resource ID for the sound for this rgn
int	the size of this rgn
binary	the region data
Str255	the text for the description box for this rgn
Boolean	the autoShow bit
Rect	the bounding rect for the description box
Rect	the link for the description box

A routine to open an interactive image is shown below:

```
/*************************************************************************/
/*                                                                       */
/*                    Boolean        openImageFile(reply)                */
/*                                                                       */
/*************************************************************************/

Boolean openImageFile(reply)

SFReply        *reply;

{
  SFTypeList    typeList;
  int         numTypes;
  Point       where;
  OSErr       err;
  int         refNum, totalRegions, theRegion;
  long        byteCount;
  RgnHandle     theRgnHdl;
  long        count;
  int         theRgnSize;
  int         window;
  Str255        tmpStr;
  Rect        tmpRect;
  Ptr         labelPtr;



  if (reply->good) {
     err = FSOpen(reply->fName, reply->vRefNum, &refNum);
     if (err == noErr) {

        count = sizeof(byteCount);
        err = FSRead(refNum, &count, (Ptr)&byteCount);

        pictHdl = (PicHandle) NewHandle((long)byteCount);
        count = byteCount;
        err = FSRead(refNum, &count, (Ptr)*pictHdl);

        if (err == noErr) {
           playSound(DRUM_SND);

           count = sizeof(Str255);
           err = FSRead(refNum, &count, (Ptr)&tmpStr);

           count = sizeof(int);
           err = FSRead(refNum, &count, (Ptr)&totalRegions);

           count = sizeof(Rect);
           err = FSRead(refNum, &count, (Ptr)&tmpRect);
```

```
displayPict(reply->fName, &tmpRect);     /* "Returns" wPtr */
addPictToImageList(&window);
   /* Uses the global var's 'wPtr' and 'pictHdl'. */

str255cpy(windowList[window].sndFileName, tmpStr);
windowList[window].ttlRegions = totalRegions;
if (totalRegions > 0)
   windowList[window].currRgnSel = 1;

for (theRegion=1; theRegion <= totalRegions; theRegion++) {

   count = sizeof(Str255);
   labelPtr = getNewPtr((long)256);
   err = FSRead(refNum, &count, labelPtr);
   windowList[window].label[theRegion] = labelPtr;

   count = sizeof(int);
   err = FSRead(refNum, &count,
      (Ptr)&(windowList[window].theSnd[theRegion]));

   count = sizeof(int);
   err = FSRead(refNum, &count, (Ptr)&theRgnSize);
   theRgnHdl = (RgnHandle) NewHandle(theRgnSize);
   if (theRgnHdl == nil) {
      displayMessage("\pCan't allocate memory needed in NewHandle.");
      ExitToShell();
   }
   windowList[window].rgnList[theRegion] = theRgnHdl;

   count = theRgnSize;
   memUsed += count;
   err = FSRead(refNum, &count, (Ptr)*theRgnHdl);

   count = sizeof(Str255);
   labelPtr = getNewPtr((long)256);
   err = FSRead(refNum, &count, labelPtr);
   windowList[window].txtWText[theRegion] = labelPtr;

   count = sizeof(Boolean);
   err = FSRead(refNum, &count,
      (Ptr)&(windowList[window].autoShow[theRegion]));

   count = sizeof(Rect);
   err = FSRead(refNum, &count,
      (Ptr)&(windowList[window].txtWBounds[theRegion]));

   count = sizeof(Rect);
   err = FSRead(refNum, &count,
      (Ptr)&(windowList[window].txtWLink[theRegion]));

   windowList[window].txtWPtr[theRegion] = nil;
   windowList[window].txtWOpen[theRegion] = FALSE;
}
err = FSClose(refNum);
windowList[window].dirty = FALSE;
```

```
            assignSoundInfo(window);
            str255cpy(windowList[window].theFileName, reply->fName);
            return(TRUE);       /* file was opened */
         }
         else {
            displayMessage("\pError reading file.");
            FSClose(refNum);
         }


      }
      else
         displayMessage("\pError opening file.");

   }
   else
      return(FALSE);            /* no file was opened */

}
```