NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC FILE COPY

DTIC
ELECTE
MAR 0 5 1991
S B D

THESIS

DESIGN AND IMPLEMENTATION OF THE ACOUSTIC
DATABASE AND ACOUSTIC TRAINER FOR "ARGOS"

by

Deborah R. Kern

June 1990

Thesis Advisor: C. Thomas Wu

Approved for public release; distribution is unlimited

91 2 28 083

Unclassified
Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification UNCLASSIFIED	1b Restrictive Markings
2a Security Classification Authority	3 Distribution Availability of Report
2b Declassification/Downgrading Schedule	Approved for public release; distribution is unlimited.

4 Performing Organization Report Number(s)		5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School	6b Office Symbol *(If Applicable)* 39	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000		7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization	8b Office Symbol *(If Applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*		10 Source of Funding Numbers

Program Element Number	Project No	Task No	Work Unit Accession No

11 Title *(Include Security Classification)* DESIGN AND IMPLEMENTATION OF THE ACOUSTIC DATABASE AND ACOUSTIC TRAINER FOR "ARGOS"

12 Personal Author(s) Kern, Deborah R.

13a Type of Report Master's Thesis	13b Time Covered From To	14 Date of Report *(year, month, day)* 1990 June	15 Page Count 106

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	Software Engineering, Event Driven Multimedia Database, Acoustic Database

19 Abstract *(continue on reverse if necessary and identify by block number)*
ARGOS is a multimedia database prototype system currently being developed by the Computer Science department at the Naval Postgraduate School in Monterey. Its primary purpose is to provide a prototype system that could be used as a Battle Group Commander's assessment tool and a shipboard data management tool. In addition to providing increased efficiency and productivity to the Navy, the ARGOS system is designed to assist in the paper reduction effort currently underway on board Navy ships. This implementation demonstrates the contribution such a system would make to the efforts of the Anti-submarine warfare (ASW) community.

20 Distribution/Availability of Abstract [X] unclassified/unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual C. Thomas Wu (408) 646-3391	22b Telephone *(Include Area code)* 52Wq	22c Office Symbol

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted   security classification of this page

i

Design and Implementation of the Acoustic Database and Acoustic Trainer
Modules for "ARGOS"

by

Deborah R. Kern
Lieutenant , United States Navy
B.S., University of Californial at Davis, 1977

Submitted in partial fulfillment of the requirements
for the degree of

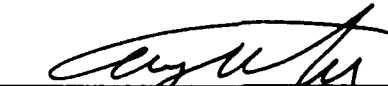# MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1990

Author: _____
Deborah R. Kern

Approved by: _____
C. Thomas Wu, Thesis Advisor

_____
Vincent Y. Lum, Second Reader

_____
Robert McGhee, Chairman,
Department of Computer Science

ii

ABSTRACT

ARGOS is a multimedia database prototype system currently being developed by the Computer Science department at the Naval Postgraduate School in Monterey, California. Its primary purpose is to provide a prototype system that could be used as a Battle Group Commander's assessment tool and a shipboard data management tool. In addition to providing increased efficiency and productivity to the Navy, the ARGOS system is designed to assist in the paper reduction effort currently underway on board Navy ships. The purpose of this thesis is to develop an acoustic database and trainer module for the operations module of the ARGOS system. This implementation demonstrates the contribution such a system would make to the efforts of the Anti-submarine warfare (ASW) community.

Accession For

NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	

By
Distribution/

Availability Codes

| Dist | Avail and/or Special |
| A-1 | |

QUALITY INSPECTED 3

TABLE OF CONTENTS

v

LIST OF FIGURES

I. INTRODUCTION

The ARGOS project is an ongoing thesis research effort being conducted at the Naval Post Graduate School at Monterey California. Initiated in January 1988, its primary purpose is to provide a prototype system that could be used as a Battle Group Commander's assessment tool and a shipboard data management tool. In addition to providing increased efficiency and productivity to the Navy, the ARGOS system is designed to assist in the paper reduction effort currently underway on board Navy ships.

The ARGOS system is a multimedia database using text, sound and graphics and is implemented on the Apple Macintosh using HyperCard. ARGOS has been divided into six functional areas; maintenance, operations, medical, supply, administration and personnel. The major emphasis of the project has been the development of the maintenance module. [Ref. 1:pp. 1-2]

The purpose of this thesis is to develop a historical acoustic database and trainer module for the operations module of the ARGOS system. This database would be utilized by acoustic analysts which make up the Anti-submarine warfare (ASW) community.

As ocean noise increases, the job of the acoustic analyst is becoming increasingly more difficult. Acoustic analysts are further hindered by inadequate and poorly organized reference material and insufficient training. Computer processed underwater sounds, known as grams, are utilized by the analysts to search for submarine data.

1

A computerized historical database of these grams would provide the acoustic analyst with a means for rapid retrieval of reference data and a tool to aid in classifying acoustic data. The training module would provide the ability to create personalized quizzes, designed to concentrate on an analyst weak areas.

Basic principles for designing a user-friendly program were used in the development of these modules. The acoustic modules are designed to provide a user-interface that will put the user in control of the system. The modules are easy to learn and provide online help for new and infrequent users. Wherever possible, the interface of the acoustic modules maintains consistency with previously developed ARGOS interface designs.

Because parts of the elements of underwater acoustics are classified, a fictitious historical data-base of whale detections was developed as a demonstration database. It consist of acoustical displays, corresponding whale migration maps, text format information of the detection, and reference documents.

The acoustic modules are designed to operate as a part of ARGOS or function as a stand alone system. The functionality of the modules could be made more powerful integrating the acoustic database with the signal processors and the communication systems. With the utilization of rule-based artificial intelligence techniques, the acoustic modules could be expanded to produce an automated analyzing and alerting system.

The goal of the acoustic modules is to demonstrate the advantages of a multimedia database of historical gram data and to serve as a prototype system in support of the design of a more sophisticated system. Although the current database is artificial, it remains an effective demonstration of how the ASW community could profit by using a multimedia database.

The organization of the thesis is as follows: Chapter II discusses the problems of the acoustic analyst in analyzing passive gram data. Chapter III is a discussion of basic principles for the creation of a user-friendly interface and how these principles were applied in the acoustic modules. Chapter IV gives a description of the acoustic database and training modules and discusses some of the design issues considered. Chapter V provides details of the implementation of the acoustic modules and provides a users -interface guide. Chapter VI describes how the acoustic database could be expanded as part of a fully integrated system and how it could evolve into an acoustic analyzer and cueing device. Chapter VII contains the conclusions of the thesis and recommendations for future thesis research.

II. THE PROBLEM STATEMENT

Since World War II, the Navy has been interested in the exploitation of sounds propagated under the water, specifically those sounds created by submarines. Anti-submarine warfare (ASW) has evolved from an obscure secret community to become an integral part of the Navy's mission and is now recognized as a top warfighting priority. Today the major method for detecting submarines is through the use of passive sonars.[Ref. 1:pp. 21-24] Analysis of passive sonar falls upon enlisted acoustic specialist who must quickly classify the sound signals. Timeliness is essential, and a matter of life and death in a wartime scenario. The analyst's job has become increasingly more difficult largely as a result of a decrease in the signal-to-noise ratio they must work with. Their efforts are furthered hindered by inadequate and poorly organized reference material and insufficient training. All these factors lead to possible miss-classifications and unacceptable time delays.

A computerized historical acoustic database would provide the analyst with a means for rapid retrieval of reference data. Additionally, this data, when used in conjunction with a training module, would provide a ready store of teaching and testing material to enhance the analyst's skill level. Incorporating the acoustic database and training module into the ARGOS project, the Navy would realize the added bonus of decreasing the amount of paper to contend with on ships.

In an attempt to passively locate threat submarines, underwater microphones, called hydrophones, are place in the water. These hydrophones pick up sound waves travelling through the ocean medium. The sounds are amplified and

4

processed through a technologically advanced signal processing method. The end result is a "picture" of underwater sounds. These pictures, known as grams, may resemble abstract art to the average lay person, but to the trained acoustic analyst, they are a window into the opaque sea. From the jumble of ocean sounds, called ambient noise, the analysts tries to find those signals which were produced by a submarine; the representation of these sounds on the gram are known as the signature of the submarine.

A key factor in the analyst's ability to "recognize" a threat signature is the amount and loudness of the oceans ambient noise as related to the sound level of the submarine's signal. This ratio of the ambient noise level compared to the signature noise level is called the signal-to-noise ratio (SNR). The lower the signal-to-noise ratio, the harder for an acoustic analyst to differentiate between a signal of a submarine and any other of the many sounds present in the ocean. This differentiation, called recognition is the key to a successful ASW mission. Poor recognition leads to miss-classifications of data and a longer time delay in reacting to the signal.

Recognition has become increasingly difficult over the last decade due primarily to a steady reduction in the signal-to-noise ratio. Two major factors that contribute to this reduction are the increase ambient noise in the ocean and the decrease in the sound propagated by Soviet submarines. The increase in ambient noise is mostly the result of increased shipping activity in the World's oceans. The decrease in signature strength is due to an aggressive submarine quieting program conducted by the Soviets.

The Soviets are believed to have been alerted to the vulnerability of their submarines by information gained through the Walker spy ring. Knowledge gained about the sensitivity of U.S. sensors convinced them that major quieting of their submarines was essential. Using legal and illegal methods , the Soviets then began to accumulate western technology to achieve that goal. Their purchase of the high precision milling machinery sold to them by Toshiba has allowed them to produce quiet propellers which has significantly increase the "stealthiness" of their submarines. In addition, the Soviets have conducted their own research and made further advances in ASW technology and strategy. [Ref. 1:p. 23]

With these factors to contend with, the job of analyzing submarines has become as much an art as it is a science and is rarely straight forward. To be a good analyst, a good knowledge base of overall submarine signatures must first be learned. In the days when the ocean was quiet and submarines were noisy, this was about all one needed to know. Now other strategies must be utilized. For example by correlating the signal being analyzed to possible actions that could be causing the signal, todays analyst tries to gain clues as to the identity of the source of the signal.

When analyzing an acoustic signature, the analysis may need to refer to past detections to use as a reference. Many ASW commands and divisions maintain a gram-cut library for this purpose. There are no guidelines or standards for the creation or maintenance of these libraries. Their organization, condition, and usefulness vary from command to command, and within commands from year to year depending on the interest level of the operators who maintains it.

Another source of reference is the classified Navy publications. These publications are bulky and often the analysts finds them too much of a bother to utilize. Those pubs that are used, soon become tattered with use. Both the official Navy pubs and the gram-cut library contribute to the problem of paper weight, storage and clutter on Navy ships.

Attempts to solve the recognition problem can be categorized into two main areas; Improving signal processing and improving the analyst's skill level. Technology can improve recognition by the development of more sensitive signal processors which will combat the effect of the reduced SNR. Utilizing Artificial Intelligence techniques to create an automatic cueing system could feasible produce a fast, highly accurate system. The development of a computer system that would continue to learn the intricacies of acoustic analysis, gaining information from mistakes as well as successes is a particularly exciting concept which should be investigated.

On the human side of improving recognition, two techniques remain the best; continual exposure to signals of interest and in lieu of that, training. Operators which have the most experience with analyzing real time submarine signals have a significantly higher recognition factor over the inexperience operator. However, in most ASW units, the analyst is required to perform many other duties and does not get the chance to continually analyze acoustic data. Th's skill of recognition decreases; the longer the operator is away from actual analyzing, the more the decrease. Training then becomes paramount in honing the skill of the acoustic analysts. Many methods are used for training meeting different degrees of success. Using the Navy publications or gram-cut logs as a training tool provides some benefits but is boring. Only the exceptionally

motivated analyst will voluntarily submit to such jejune lessons. Often times the training consist of the same gram-cuts which eventually are memorized by the analysts. Once this occurs, the training is not effective. The analysts is given a false belief in his analyzing skills which will fall short when confronted with an unfamiliar signature. Running a previously taped signal through the signal processor gives the analysts the most realistic training but interferes with real-time processing and so can not be used on a regular basis.

The goal of this thesis is to demonstrate a computerized approach which, if implemented, would contribute to the solution of the recognition and training problem. Two modules have been developed as part of the operations section of the ARGOS project. One is a historical database and the other is an acoustic trainer. In keeping with the effort to create a "paperless ship" navy, these modules have been designed to replace publications and references that currently clutter Navy ships.

A historical acoustic database would provide the acoustic analyst the means for rapid retrieval of reference data and a ready tool to aid in classifying acoustic signatures. The analysts will no longer have to flip through pages of data, looking for an example of a specific frequency, source or submarine type but can simply query the database and receive the needed information. This will result in quicker recognition of signal data with fewer miss-classifications.

The training module would provide an easy means to create personalized quizzes designed to concentrate on an analysts weak areas. By using the acoustical database, it would have a rich supply of grams and reference material to choose from. Maintaining the training system apart from the signal processor

8

would allow training to occur at anytime, resulting in more training being conducted. By creating a friendly user interface, the analyst will be encouraged to use the system as a tutorial and work individually on the system.

This thesis is offered as a partial solution to the recognition and training problems discussed above. Research must continue to produce increased sensitivity in the signal processors and the incorporation of artificial intelligence cueing and training devices.

III. USER-MACHINE INTERFACE

A. INTRODUCTION

The design of the user-machine interface is a critical step in the development of any software package. It is through the interface that a user can be either helped or hindered to accomplish a task. A user friendly interface will produce a system that is easier and faster to learn. The user will spend less time participating in costly training and quickly begin productive work. The system with a good interface will decrease the amount of errors made by the user. In addition, such a system will be easier for the user to remember how to operate. Thus, the occasional user will not have to waste time relearning the system every time he wants to use it. The development of a user friendly interface can result in higher productivity of both the user and the software, and reduced cost due to training , and time wasted on errors. This chapter discusses basic interface principles and some of the factors to consider when designing the user interface.

B. FACTORS THAT INFLUENCE USER INTERFACE DESIGN

The design of a system should be geared toward the type of people that are going to be using it. While the diversity of people will make it impossible to design a system that pleases everybody, studying a profile of the user group will allow a system that meets approval to the widest base possible. Some factors to be considered are discussed below.

1. The User's Knowledge Level

Although the use of computers is growing, the percentage of people who are experienced with computer operations is still relatively small. The interface needs to be designed in a way that will best suit the intended user. A system that will be used by novices will need to be designed differently than one which is used by experts. Systems that will be used by individuals with varying skill levels should be designed to provide ease of use to the beginner and shortcuts for the experienced user. [Ref. 3:p. 51]

2. Personality Of The User

Age, sex, physical abilities, education, cultural or ethnic background, training, motivation, goals and personality all play a part in how a user will respond to various designs. [Ref. 4:p. 53] For example, using a color code to indicate different functionality will be useless if the operator is color blind. Some preferences can be determined by understanding the community of users the system is for. A military users group will prefer time to be represented using the 24 hour time system, while a system for a small business most likely will use the 12 hour system.

3. The Intended Use Of The System

How and where the system will be used can also affect the interface design. For example, if the system is to be used in an area of high noise, then using a bell or other sound device to alert the user to an error will not be effective.

C. PRINCIPLES OF USER INTERFACE

1. Putting The User In Control

A user will feel in control of a system if the actions that must be performed make sense. All users have a semantic model of the tasks needed to perform their job. A system that matches this model will allow the user to predict the actions needed to perform an operation. Many of the principles that follow contribute to putting the user in control. [Ref. 3: p.50]

2. Consistency Of Data Display

Consistency in a program makes a program easier to learn and easier to remember once learned. One method humans use to remember is by rehearsal; repeating something over and over until it becomes a normal response. Consistency in a program contributes to the rehearsal process.

3. Graceful Way to Recover From Errors

Any system that is operated by humans will eventually have to deal with incorrect input. A good system will provide feedback to the user that is understandable and provide help in recovering from the error. A system that crashes when because of an error will intimidate many users and frustrate all the users.

4. Minimal Memory Load on the User

It is easier for most users to recognize the command they want from a list of choices than to recall it from memory. By reducing the amount of commands or actions required, the user can concentrate on the job at hand and not controlling the system.[Ref. 3:p. 55]

5. Provide Online Documentation

Online documentation includes messages that inform the user of the status of the system, error messages and prompts, as well as detailed and quick reference help. The online documentation help the user to feel in control of the system. [Ref. 3:p. 54]

6. Protect User From the Inner Workings of the Computer

The interface should be designed to take input in a way that makes sense to the user and then convert it to a format that is usable to the computer. The user should not be subjected to phrases or error messages generated from the internal system. This type of feedback will only confuse and frustrate the user and provide no useful information as to how to correct the error. [Ref. 3:p. 53]

D. SUMMARY

Of course, a good user interface will not make a productive system out of poorly designed software , but it does have a major role in the effectiveness of any system. These basic principles of user interface design have been considered in the design of the acoustic modules and will be described in the next chapter.

IV. DESCRIPTION OF ACOUSTIC MODULES

A. INTRODUCTION

As discussed in the previous chapter, creating a program environment that is easy to learn and use will increase the productivity of the user. This chapter describes the user interface of the acoustic modules and discusses how the design principles of chapter III were incorporated into these modules.

B. ICONS

The Acoustic modules were designed to be consistent with the design previously used in the ARGOS project. In considering the use of icons, a concentrated effort was made to ensure that each icon was self explanatory. A help function exist to explain each icon in case this effort failed. Whenever possible, icons which are used in other modules of ARGOS are also utilized in the acoustic modules. Functions unique to the acoustic modules use icons or buttons that are different enough from other ARGOS standard icons so as not to confuse the user. Both modules are designed to be used as a part of ARGOS. However, the modules are also complete in themselves and could be used as a stand alone system.

C. THE INTENDED USERS

The expected users of these modules are enlisted analysts with analyzing skills and experience levels which ranges from novice to expert. As acoustic analysts, some assumptions about their general computer aptitude were made. Before enlisting in the Navy, each sailor is given an aptitude test designed to determine individual strong and weak areas. To qualify for the analyst rating, a

14

sailor must achieve high scores in the math and computation areas. Therefore it is assumed they will have an aptitude for working with computers. In addition, they will have gone through 'A' school where they will have received extensive instruction and hands on experience in operating specialized computer systems. The modules have therefore been designed for the user who has performed operations on a computer of some type before and does not have to overcome 'computer phobia'.

D. ROLE OF THE ACOUSTIC DATABASE

The acoustic database is intended to be used in a real-time classification scenario. Several scenarios exist and can be divided into two main area: using the acoustic database as a 'heads up' cueing device and using the database as an aid to classifying a 'real-time' target of interest. The following sections discuss these two types of scenarios.

1. Heads Up

In this scenario, the analysts has advance notice of a particular type of submarine to look for, but the submarine has not yet been detected by any ASW sensors. Using the acoustic database, the analyst can retrieve gram and message data from past detections of this class of submarine and review these signatures. This review will sensitize the analyst to the most likely signatures. This has the effect of increasing their recognition differential and will produce a quicker detection.

In another scenario within the heads up area, the analysts has advanced notice of a particular type of submarine which has been detected by some ASW sensor. In this case, the analysts on the platform that detected the unit could refer to the acoustic database and reference the best resemblance of that unit and

15

pass that reference to other ASW units. Upon receiving the reference, the analysts could then retrieve that particular gram to review. If no gram reference is given, the analyst can determine the detection frequencies from the real-time message data and can retrieve data on those types of submarines that were detected on those frequencies.

2. A Classification Aid

In this scenario, an analyst sees something on the gram and is unsure if the sound source is generated from a submarine. By retrieving all data on the line or lines of interest, the analysts can compare past detections with what he is looking at. This comparison will assist the analyst to make the initial classification of submarine or non-submarine and if it is a submarine, assist to further refine the classification. The acoustic database does not have to be restricted to only 'real' signatures. Confusing gram data that is similar to a submarine signature and may have fooled or confused analyst before can be included so the mistake is not repeated.

E. INTENDED USE OF THE ACOUSTIC TRAINER

The acoustic trainer can be used as either an informal or official testing and teaching device. As an informal trainer, anyone is allowed to make a quiz and creating a quiz is a simple task. The informal trainer will hopefully encourage friendly competition between analysts or sections with no official scoring taking place. This will remove the pressure of a grade from the learning process and will hopefully promote a more productive learning atmosphere.

The trainer can also be used for official qualification testing. Security of these test could be done by password protection, however, the simplest method would be to keep the test data on a disk and physically secured.

16

F. MODULE ACCESS

As a part of ARGOS, the acoustic modules can be accessed by selecting the desired ship on the top level ARGOS card. This selection will take you to the Functional Main menu of ARGOS and list the six functional areas of ARGOS. Selecting the Operations button will provide a menu of options within the operations area of responsibility. By highlighting the acoustic database option, the user will reach the main entry point into the acoustic modules (Figure 1). If the modules are used as a stand alone system, this menu will be the first card accessed when opening the system.

Access through the acoustic database is organized in a hierarchical format. Initial entry can only be reached through the main menu. Once in the module, the user has a defined path that he is allowed to take. The intention of limiting the access paths between cards is to prevent the user from getting lost in the database. If the user ever does feel lost or needs to quickly exit from the system, he can return to the acoustic main menu by selecting the lighthouse icon from any card in the acoustic database. The hierarchical design of menus was used to support the user-interface principles of putting the user in control and creating an environment that is easy to learn.

G. THE ACOUSTIC DATABASE

Figure 1 shows the main menu for the acoustic modules. This menu provides the user with the options to retrieve specific data, browse through thedatabase or enter the acoustic training module. Selecting the lighthouse icon

while in the acoustic main menu card will take the user back to the ARGOS main menu. The help icon provides a brief description of the the three choices provided by the main menu.
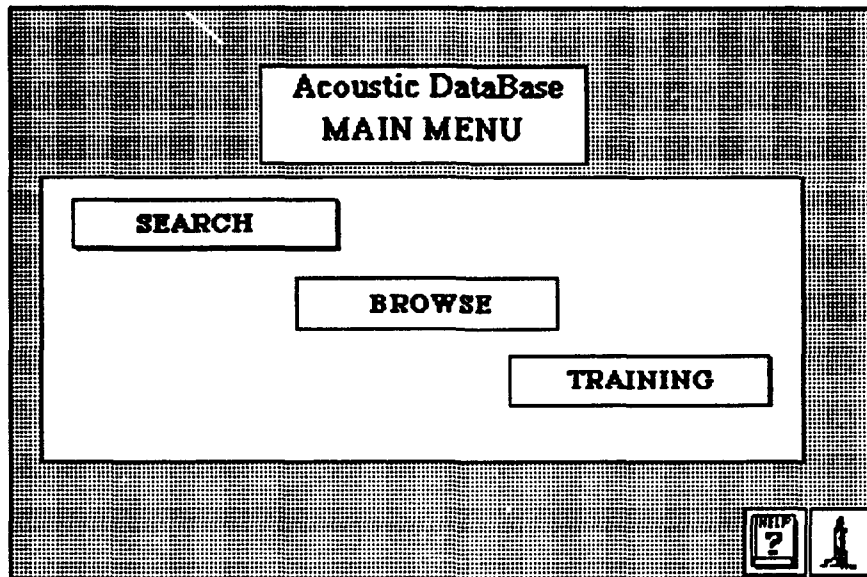
Figure 1 Main Menu

1. Search

Clicking on the Search button (Figure 2) will give the user the choice to retrieve specific data by selecting the desired parameters for search or retrieving the entire database. Choosing the Selected Search option will cause the query builder card to be displayed (Figure 3).

Figure 2 Main Menu Search Options

18

The user can request a search by name, type, id, date of detection, detected cycles, detected sources, location, or other. The search by location and other functions have not been implemented at this time. When implemented, location will provide options for retrieval of data located in a specific region of the ocean such as Gulf of Alaska. The 'other 'option will be for a search of a key word in the narrative portion of the summary message.

Placing the mouse cursor on any of the buttons along the top of the screen (Figure 3), will provide a pull-down menu with selections available for that field (Figure 4). When the user clicks on a highlighted selection, the selected item will appear in the corresponding field in the lower half of the card.

By providing all selections available in a menu, the user can have no doubt of the data that needs to be entered or the format required. This will reduce the potential of errors due to typing or spelling mistakes. In addition, the user will not have to rely on memory for the command options available. This will reduce the amount of reliance on long term memory and thus shorten the learning curve and increase the retention period for the use of the system.

NAME	Type	ID	Date	Cycles	Source	Location	Other

Selected Parameters for Search
Find cards that meet all requirements

Name:

Type:

Location:

Source

Date:

Id:

Word
Search:

Cycles:

```
HELP
 ?
```

Figure 3 Query Builder

NAME	Type	id

```
Beluqua
Blue
Finback
Humpback
Orca
Sei
Sperm
Right
```

Figure 4 Pull-Down Menu Selections

Multiple selections of an individual field are allowed in those fields where more than one value is possible. For example, since it is possible for a detection to consist of more than one source, the user is allowed to select multiple choices from the source field. However, it is not possible for a whale to be classified as both a Humpback and a Sperm. If a user selects another choice in the type field , a dialog box will appear (Figure 5) informing the user that only

20

one choice is allowable. The user is then given the option of replacing the old selection with the new one or canceling the request.
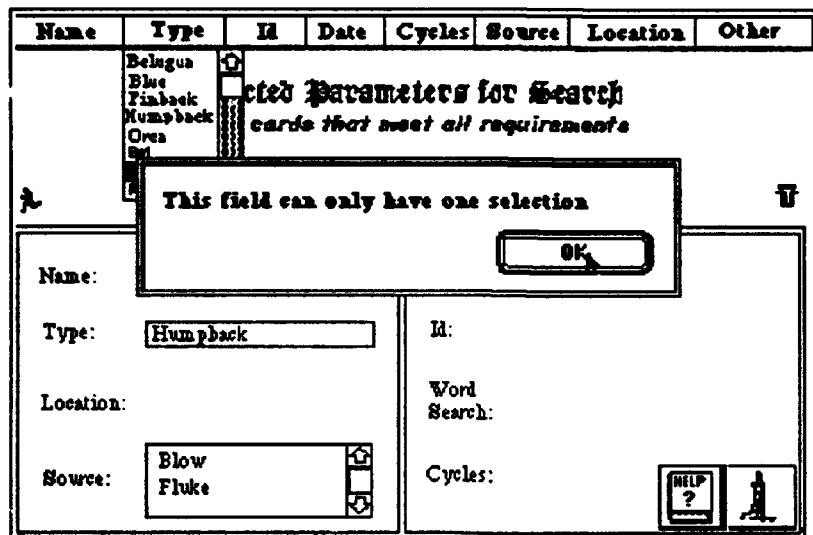
Figure 5 Query Builder With Error Message

The trash can icon will clear all the fields and present the user with a 'clean' card ready for a new query request. The trash can was positioned apart from other icons to prevent it from being selected by mistake. In particular, it was place away from the execute icon, since those two commands are at opposite semantic meanings.

The data search is executed by selecting the icon of the running man. The summary messages will be searched for data that meets the requirements specified by the user and its data is used to generate a report (Figure 6) which includes the name, type, id, detected sources and cycles, date and location of each whale that met the specified parameters.

name	type	id	source	cys	date	location
whale 7-85	Humpback	111	Fluke	4.0	14 Feb 85-15 Mar 85	Gulf of Alaska
			Blow	5.0		
whale 12-85	Humpback	115	Fluke	4.0	10 Apr 86-21 Apr 86	San Francisco Bay
			Phonation	4.3		
			Blow	5.0		
whale 1-87	Humpback	111	Breach	1.8	12 Feb 87-03 Mar 87	Gulf of Alaska
			Fluke	4.0		
			Blow	5.0		
whale 9-87	Humpback	111	Breach	1.8	07 Apr 87-21 Apr 87	Sea of Cortez
			Fluke	4.0		
			Blow	5.0		

Figure 6 Query Report Form

After viewing the report, the user may return to the search card to perform an additional search by selecting the menu icon or return to the acoustic main menu by selecting the lighthouse icon. By clicking anywhere in the name column, the user is taken to a temporary stack that contains copies of the summary messages that correspond to the report. The user may browse through this mini-database in the same way as the with the main database. Clicking the trash can icon will delete the record after verifying that deletion is indeed desired.

The retrieval described above was an *and* operation. To perform an *or* operation, the user must conduct multiple searches. For example, if the user wants to retrieve all detections of breach or flukes sources, a search would be requested on either breach or fluke detections only. When the search is completed, the user would return to the search menu without deleting the existing report and conduct a search of the source not yet retrieved.

When the execute button is selected the program will check to see if the report card is empty. If there is data on the report then a pop-up window will appear giving the user the option to delete the old data, add the new data to the bottom of the existing report, or save the current report data in a new file. By selecting the add button, the data on the second search will be added to the report on breach detections.

Two problems exists with this method of conducting an *or* operation; First,the steps to conduct this search are not intuitively obvious to the user. Secondly, duplication of records will exist for those records that meet the requirements of both of the *or* searches. The following paragraph discusses the reasoning for this not so intuitive method. The second issue will be discussed in Chapter V.

Originally, the acoustic module provided a separate selection for *or* operations. However, the execution time for the data search was very large. It was felt that the slowness of the system would lead to more frustration of the user than the round about method described above. Online help will provide the user with enough information so that the learning rate will still be maintained at an acceptable level. It is recognised that this is not the optimal solution and future modifications will hopefully address this problem.

2. Browse

The browse mode is provided to allow the user to browse through the database. By selecting the browse button, the options to go to grams or summary message is provided (Figure 7). If grams is selected, the user is taken to the first card in the grams stack (Figure 8).

Figure 7 Main Menu Browse Options

This option was provided because many acoustic analysts enjoy randomly looking through acoustic grams and seeing how quickly they can correctly analyze it. Others just want to refresh their memories to as many grams as possible.
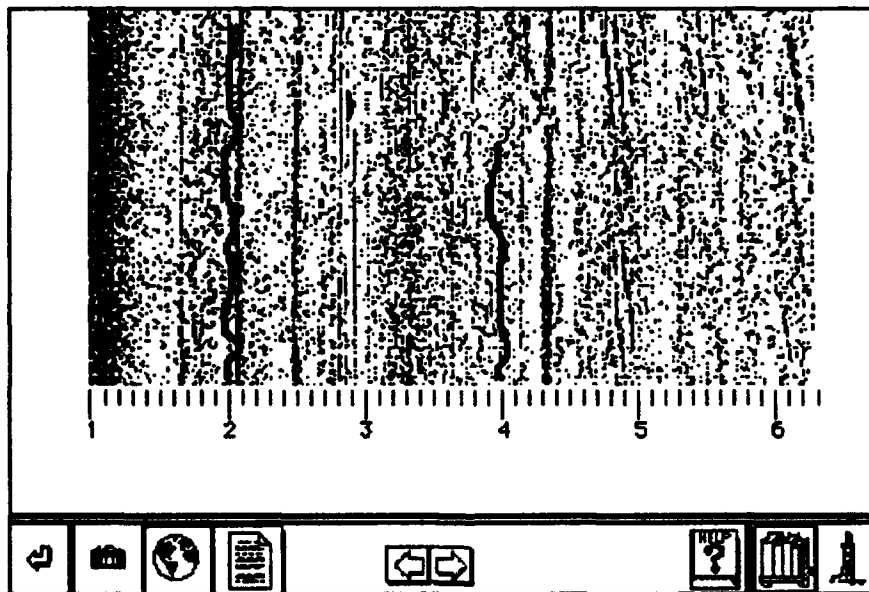
Figure 8 Acoustic Gram

The user can go through the database by selecting the right and left arrow buttons. The left arrow button will take the user to the card physically located before it in the stack. If the user is at the first card of the stack then this button will go to the last card of the stack. The right arrow button goes to the

24

next card in the stack, if at the last card of the stack then it will go to the first card. The bent arrow will return the user to the last card visited prior to the current card. The camera icon will go to a picture that corresponds to the type of whale the current gram represents (Figure 9). The globe will display a map (Figure 10) with the whale track.
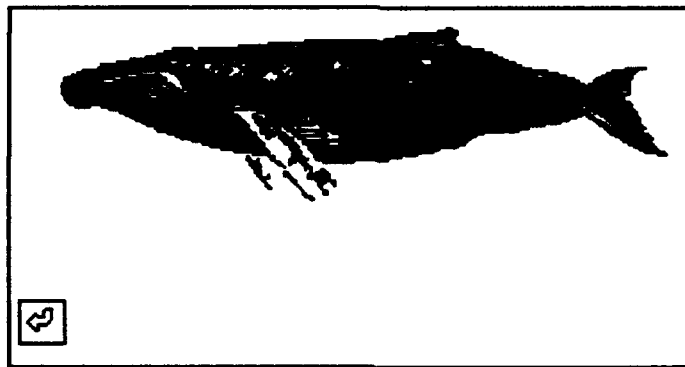
Figure 9 Picture Card

The only option from the picture and map stack is to return to the previous gram, thus restricting the access path and making it harder to get lost. Going back to Figure 8,the summary icon ,which looks like a sheet of paper, will take you to the summary message stack. From this stack (Figure 11), you have the same ability as if you selected summary messages from the browse mode.
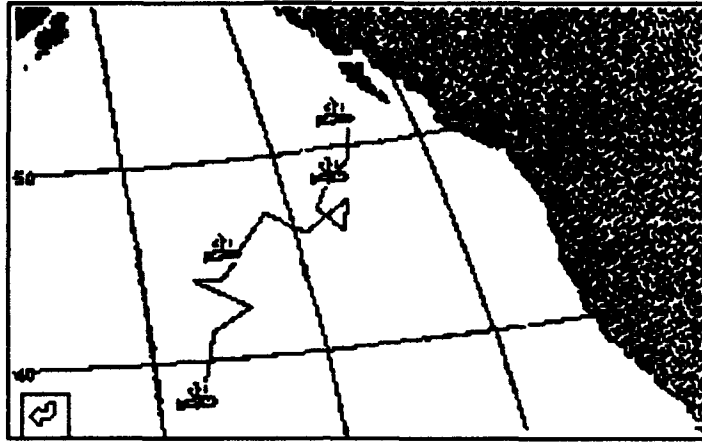
Figure 10 Track Card

Although this deviates from our principle of keeping only one path through the program, I believe that the ability to jump between the gram and message data is a good semantic model of how gram analysis is accomplished and will not confuse the user. In fact, the lack of this ability would most likely result in complaints.
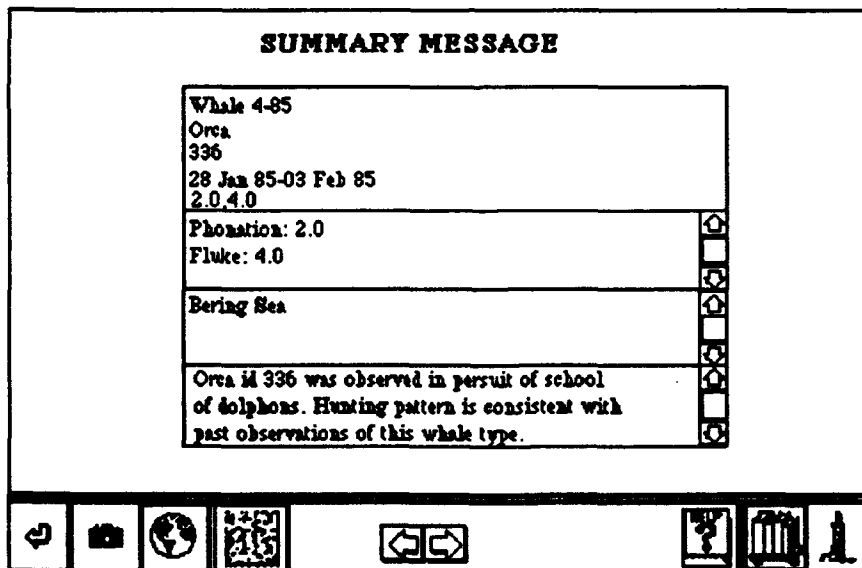
Figure 11 Summary Message Card

The summary message stack provides the text data on the detected whale. All icons are the same as previously discussed in the gram stack with the exception of the gram icon. This icon is located in the same physical screen position as the summary icon is positioned in the gram stack. Its function is to take the user to the gram corresponding to the summary text data.

Consistency has been maintained by maintaining the same icons and the same positions between the two stacks. This is especially important with the summary message and gram data stacks because of the close relationship between the two. It is anticipated that these two stacks will almost always be used in conjunction with each other.

H TRAINING MODULE

Figure 12 shows the acoustic main menu while selecting the Training button. The user is given the option of entering the training module in one of two modes: the student mode or the instructor mode.

Figure 12 Main Menu Training Options

If the instructor mode is selected, then a password will be requested (Figure 13). If the user enters the correct password then the training module is opened in the instructor mode. This mode allows the user to create or delete a quiz or a tutorial, preview a quiz or tutorial, and review student answers to

27

quizzes. An incorrect password entry will cause an "invalid password" prompt message to appear and return the user to the Main Menu.
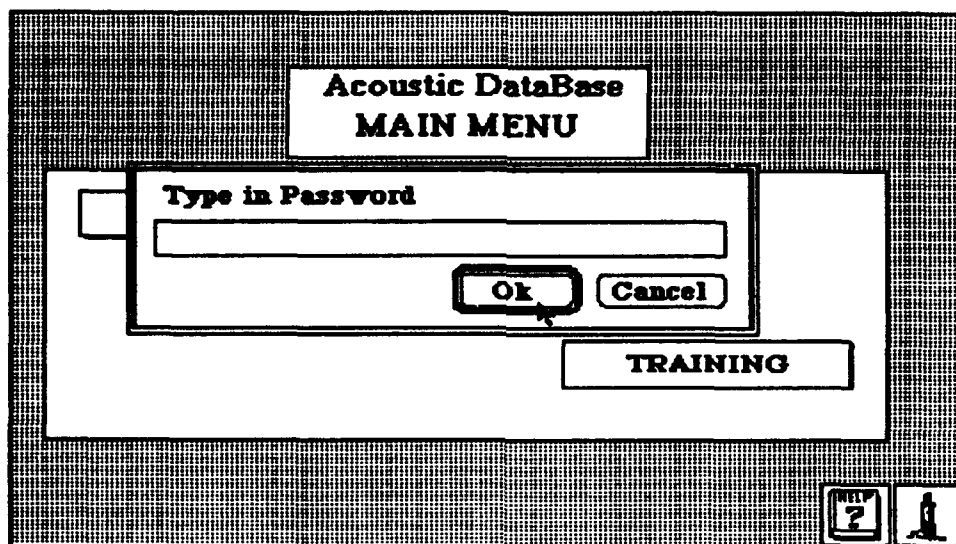
Figure 13 Password Request Dialog Box

A quiz consists of some number of grams to be analyzed and questions concerning the gram. To create a quiz, (Figure 14) the create button is selected. The user is given the option of creating a quiz or a tutorial. Upon selecting the quiz, a dialog box will prompt the user to enter a name for the quiz. Once a name has been selected, the quiz constructor card is displayed.

The quiz constructor card, (Figure 15) consists of two empty fields; The first is titled "Grams for Quiz XXXXXXX" and is pre-filled with the name of the quiz just assigned by the user. To select grams to be included in the quiz, the cursor is moved into the "Select Test Grams" button. This will cause a scrolling field with all available grams for selection. Grams are selected by highlighting and clicking on the whale name. The selected whale will be assigned a sequential

gram number and appear in the "grams for quiz ..." field. Similarly, questions can be selected from the quiz questions button. These questions will appear in the "quiz questions" field.
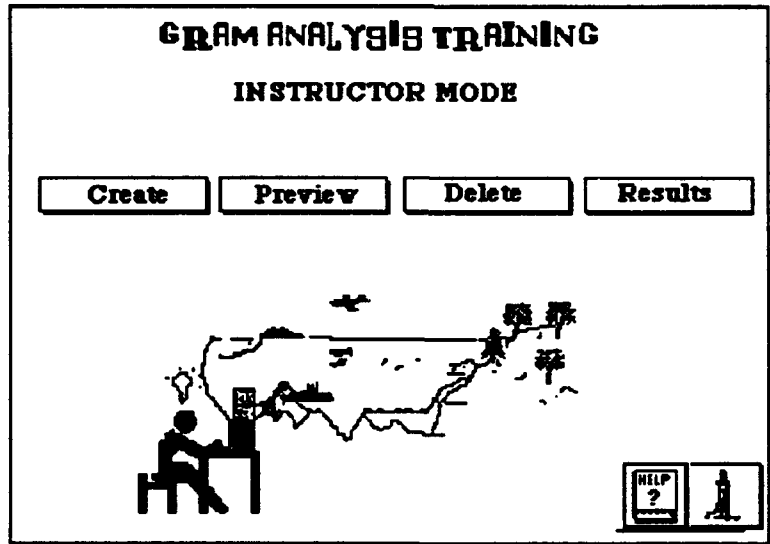
Figure 14 Training Menu (Instructor Mode)
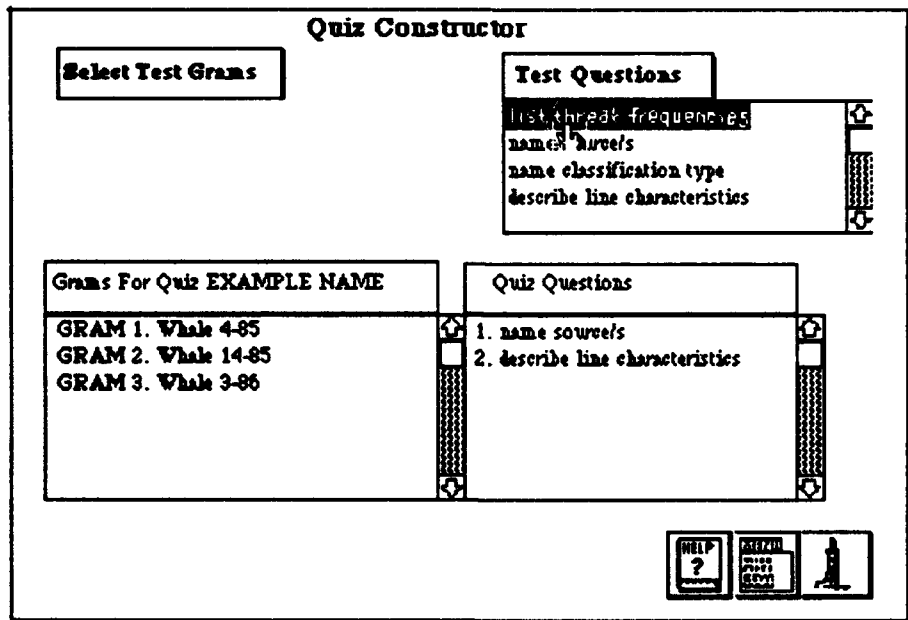
Figure 15 Quiz Constructor Card

The actions required to create a quiz are similar to those required to request a data search discussed earlier. Also the design of these cards both have the selection items at the top and the fields that display the selections in the bottom half. This consistency will reinforce the learning already achieved by learning the other function. The reinforcement of actions, plus the fact that fewer different actions are needed to be memorized will allow the user to retain the knowledge to run the system for a longer time.

Menus have the advantage of allowing the user to see the options available and help prevent errors. A disadvantage with menu driven system is that some flexibility can be lost. This could hinder the creativity of the user. With this in mind, the quiz question field has the added feature of allowing questions to be entered directly into the field. This will allow the user creating a quiz to ask questions not included in the menu selections. Since the data entered into this field does not have to be deciphered by the computer, it was felt that possible typing errors made by the user would not severely degrade the system. While a misspelled word can totally baffle a computer, the user will be able to decipher the question the majority of the time.

When the quiz is completed, the menu icon is selected and the user is returned to the main driver of the trainer mode. Moving the cursor to the delete quiz or preview quiz button will display all the quiz names currently existing. To preview the quiz, the user selects the quiz name desired and clicks. The user can now review all the grams and questions in a mode similar to the student. While in the preview mode, the user may return to the quiz constructor card for the quiz currently being previewed and make changes to the quiz.

To delete a quiz, simply move the cursor to the delete button and select the desired quiz from the scrolling field that appears. As with all actions that deletes a file in this module, the user is asked to verify that a deletion is really desired.

The results button displays the quiz results card (Figure 16). This card contains a list of quiz names, the name of student taking the quiz. Clicking anywhere on the line of the desired selection will take the user to a record of questions and the student's answers to the quiz (Figure 17). The instructor can review all answers for grading. The return button will bring the user back to the top level of quiz review. From this card, the menu icon will take the user back to the training menu.


```
                         QUIZ RESULTS

┌─────────────────────────────────────────────────────┐ ⬆
│ Quiz readytest1 results for Sailor A                 │
│ Quiz readytest1 results for Sailor B                 │
│ Quiz EXAMPLE NAME results for Sailor B               │
│ Quiz EXAMPLE NAME results for Sailor D               │
│ Quiz readytest1 results for Sailor A1                │
│                                                      │
│                                                      │ ⬇
└─────────────────────────────────────────────────────┘
```
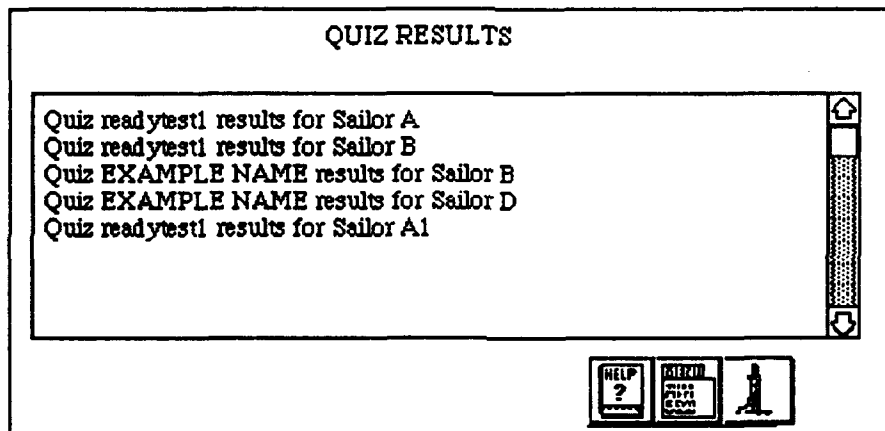
Figure 16 Quiz Results Card
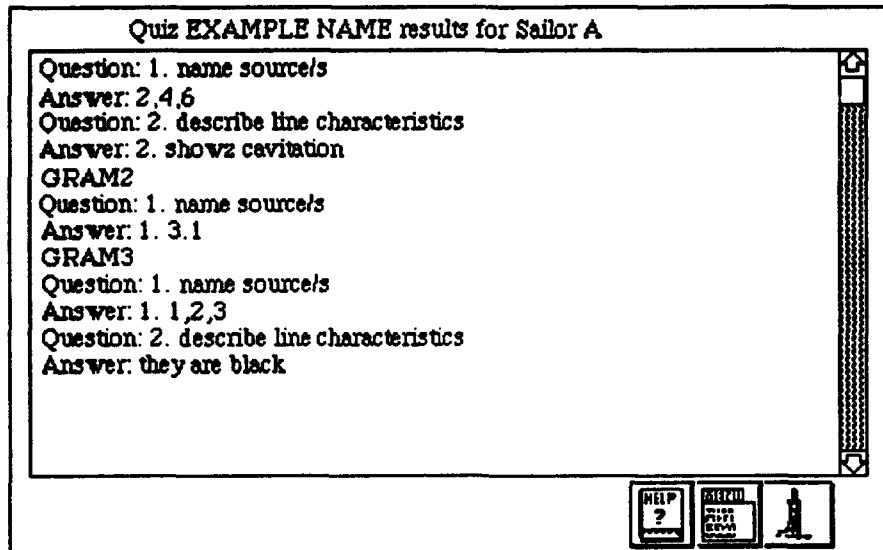
Figure 17 Student Answers Record Card

Creating a tutorial is similar to creating a quiz. The Tutorial is more restrictive in that the user can use only the questions which exist in the questions field. The user can not edit the question field as is allowed in the quiz mode because the answers and hints to the questions are pre- filled in the gram cards. Tutorials do not keep a record of student responses an therefor are not includes in the Quiz Results Card.

If, from the main menu, the student mode is selected, a training card with fewer options is opened. (Figure 18) The student will be able to select a quiz or a tutorial. If the quiz is selected, a dialog box will prompt for the name of the user. Once that is entered, the first gram of the selected quiz (Figure 19) is displayed.
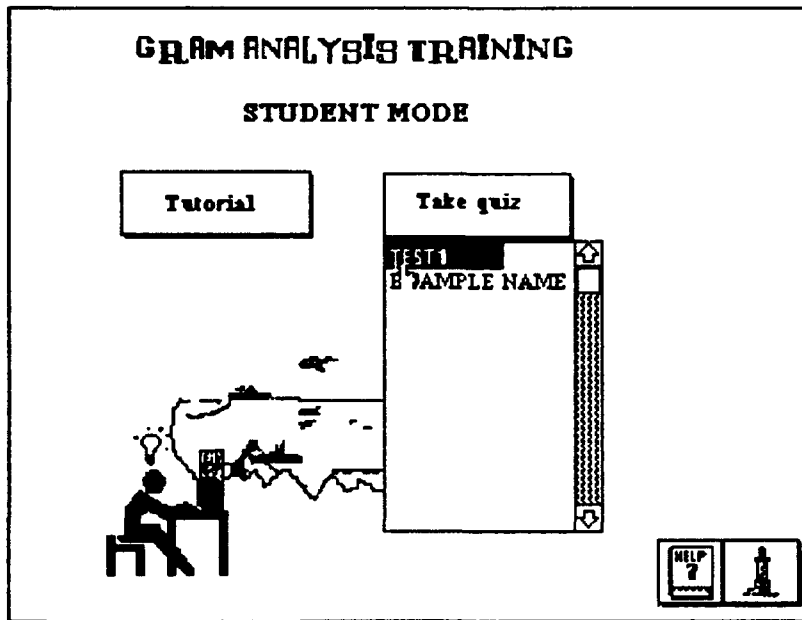
Figure 18 Training Menu (Student Mode)

1. list threat frequencies
2. Gram 1 only: what property is the line at 3.2 showing?
3. name classification type

End Quiz
Next Gram
Show Status
Hide Questions   Gram 1
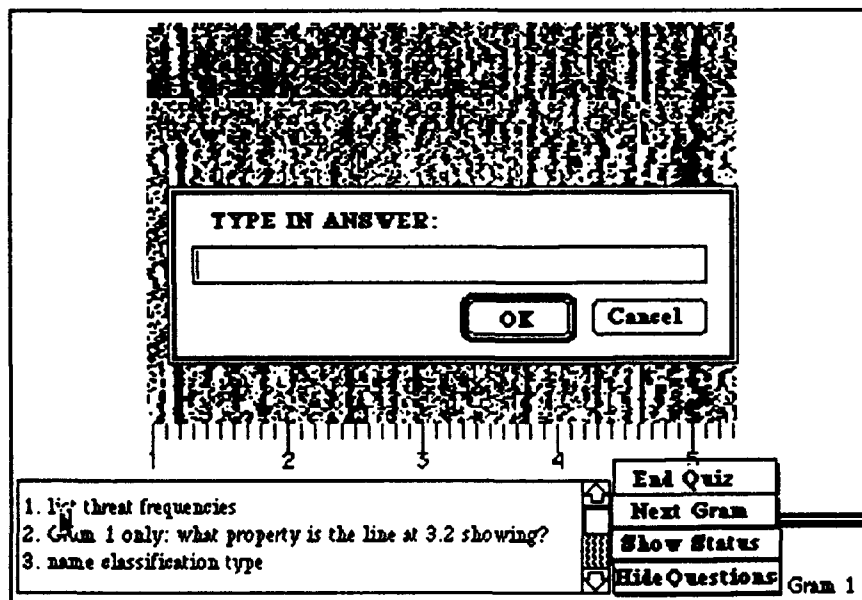
Figure 19 Gram Quiz With Answer Box

The design of this card had to deal with a major constraint. The size of the acoustic data could not be made any smaller without interfering with the users ability to analyze the gram. Therefore, very little room was available for questions or other buttons. To conserve the space available, the show question and show status buttons were designed. The show question button, when selected will display the questions for the quiz. When the questions are displayed, as in Figure 18, the show question button is replaced with a hide questions button. The show status button behaves in a like manner.

To answer a question, the user double clicks on the desired question. A pop-up window will appear for the user to enter the answer. The box disappears when either the ok or cancel button is selected or when the return key is pressed. If the user wishes to review his responses to the questions, he can select the show Status button (Figure 20). A record of the users responses is listed for review. As implemented, the user can edit any part of this field. This decision was made with the thought that most exams allow the test taker to have scrap paper or at the very least, an eraser. Knowing that every key stroke is being recorded may cause some uneasiness to the student and thus hinder the learning process. Having the ability to review the quiz record exactly as the instructor will see it may make the student a little more at ease.
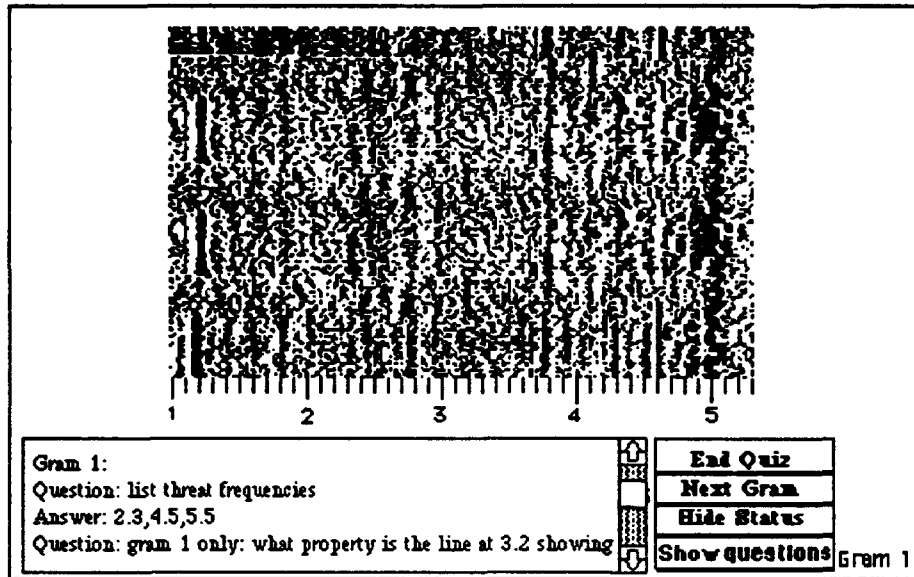
Figure 20 Gram Quiz with Answer Status Displayed

The Next Gram button will take the student to the next gram of the quiz. The gram number will always be displayed on the lower left corner. This test gram number is designed to help the student keep track of where he is in the quiz. If the next gram button is selected while at the last gram of the quiz, the first gram of the quiz will be displayed.

If the Student is in the tutorial mode, the functionality is basically the same. In the tutorial mode, a running script is no longer maintained and answers are not inputted to the computer. When the student wants to see the answer to a question, he clicks on the desired question as in the quiz mode. Now instead of a window for the user to enter the answer, the user is displayed a box with the answers (Figure 21). The hint button when implemented, will give the user some guidance toward answering the questions without revealing the answer.
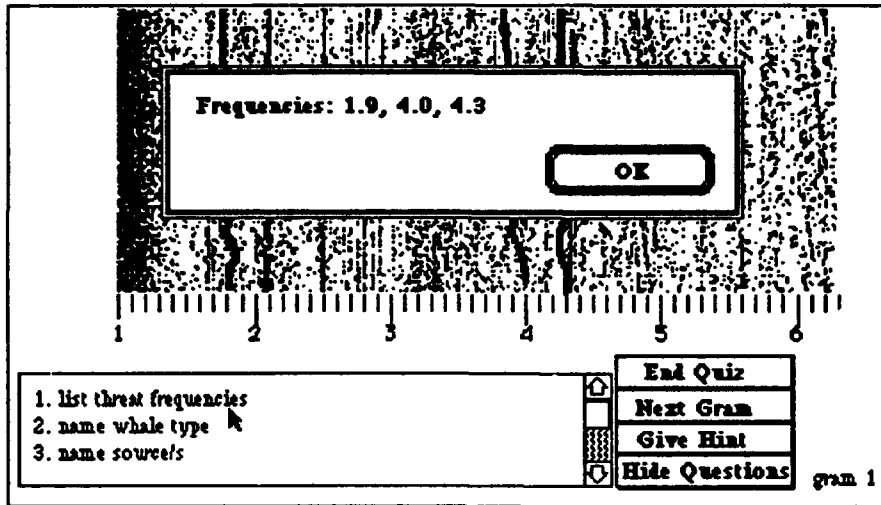
Figure 21 Gram Tutorial Card

Selecting the End Quiz button will cause a dialog box to appear which will verify that the user really wants to quit the quiz. If the user answers yes, then the quiz will end and the user will be taken to the top main menu of the acoustic modules.

I. HELP

Each card contains a help function which provides varying degrees of help depending on how complex the card operations are. The help function as a minimum will contain an explanation of the buttons and icons that can currently be used and directions on how to quit or escape the card and return to the main menu. Figure 22 show the help card associated with the Summary Card. To access the help stack, the user selects the help icon on any card. The help card that will then be displayed will be dependant on which card the help was called from. Once in the help card, the user can access the specific help needed by

36

clicking on the icon. Only those icons that are in the card from which help was called are displayed.

HELP CARD: Click on the icon to get information about that icon's function

The gram button will take you to the gram stack showing you the gram of the whale specified in the summary message.

quit help
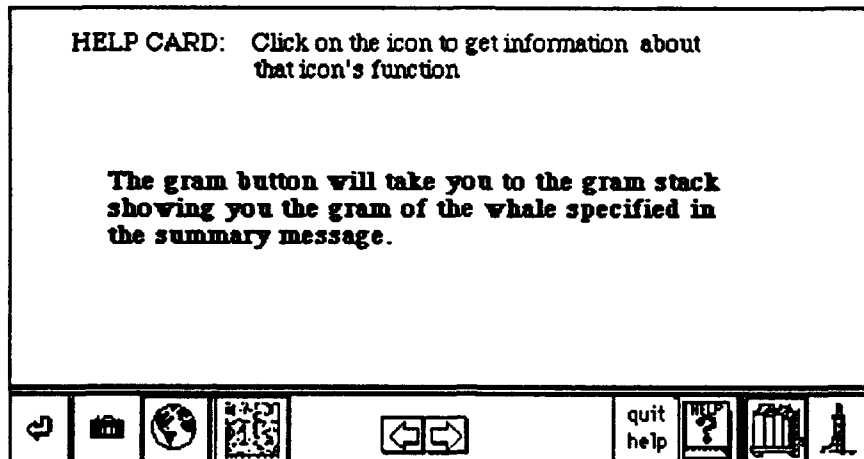
Figure 22 Help Card For Summary Stack

J. SUMMARY

When designing the acoustic modules, a conscious effort was made to follow the basic principles for creating a user-friendly interface. Icons and menus were used to reduce the amount of memorization required to use the system. On line help and documentation is available to the user at any level in the modules. A clearly defined pathway through the module was created to prevent the user from getting lost in the modules. By maintaining consistency with the rest of the ARGOS system, it is expected that the acoustic modules, if implemented would require little training time before they could be productively utilized.

V. DESIGN OF ACOUSTIC MODULES

To maintain consistency, many of the design decisions made by past contributors of the ARGOS project have been applied to the acoustic modules. One such choice was the utilization of HyperCard as the developmental environment. HyperCard as a developmental tool, has many positive aspects. Decisions on the organization of the acoustic modules into separate stacks were focused on trying to design a system that could be easily modified or extended. Design decisions with respect to the data retrieval need to be rethought and are discussed below. A major constraint in the design of the acoustic modules was the desire to restrict all aspects of the modules to an unclassified level. Since a fully functional model would be classified, it was decided that the acoustic modules should be developed as an unclassified prototype system.

A. THE PROGRAMING ENVIRONMENT

The HyperCard environment provides a powerful tool for generating user interfaces. Its ability to link differing data types makes it ideal for the development of a multimedia database, plus, the ease in making quick changes to the user interface makes this an ideal environment for rapid prototyping. HyperCard is loosely based on object oriented programing and as such deals with the manipulation of objects instead of data. Unlike other object oriented languages, the objects in HyperCard are fixed. HyperTalk is the programming language of HyperCard and is a very high level language and extremely easy to understand. As with any language that emphasizes readability, some optimization can be lost in certain circumstances. This is overcome in HyperCard with its

extended capabilities through the external command (XCMD) and external function (XFCN) code resources. This allows the developer to use or create external routines that can be written in any Macintosh compatible language, such as C or Pascal. [Ref. 5:p. 521]

The diverse types of data involved in the acoustic module fits nicely into the concepts of HyperCard's fixed objects. Items that are similar, such as gram data, can be placed in a single stack in individual cards. They can inherit common properties from the stack or background level and maintain their own individual differences at the card level. The linking of gram data to the textual information or any other media source, can be easily accomplished using HyperCard.

B. ACOUSTIC MODULES

The data objects for the acoustic database consist of acoustic grams, text messages, pictures, and maps of tracks. The gram consists of an acoustic gram paper display that was scanned to the computer. The text message is a formatted text which contains information about a specific 'whale detection'. This information consist of the whale assigned name, the whale type, id, frequencies that were detected and the source of the sound, dates and areas of detection and a free form narrative. The picture of the whales were taken from a book about whales and scanned into the computer [Ref. 6:pp. 15-100]. The map and tracking data is a simplistic representation of the whales migration. This area was not emphasized because of existing powerful localization and tracking systems currently being used by the Navy.

Unlike some of the other ARGOS modules, the acoustic modules were designed as a demonstration system for rapid prototyping. As a prototype system the acoustic modules could be demonstrated to acoustic analysts for their

input. In many cases, it would be possible to incorporate suggestions made by the analysts on the spot. This allow the analysts immediate feedback and also insures that the developer understands the suggestions made. Using this development method would eventually produce a system that would be efficient and meet the users needs. In addition, the analysts would be more favorable to the new system, since they had a major say in how it was developed. The positive attitude of the analysts toward any new system is a major key to the eventual success and productivity generated from it.

1. Organization Of Module Stacks

A useful feature of HyperCard is the ability to separate data into different stacks. This makes it simple to design modularity into a program. Changes can be made to individual stacks or cards without affecting the other stacks. For example, changing the format of the data query report form would be done entirely within the data query card.

With the goal of creating a system that can be easily modified or expanded, the acoustic modules have been divided into eight stacks, grouping similar data types and functionality together. The database is divided into four stacks; one holds only gram data, a second stores textual summary message data, the third contains pictures and fourth has the map data. The remaining four stacks contain the menu cards, quiz cards, the query reports, and the quiz reports.

2. Data Query

Chapter IV described the process for achieving an *or* operation by merging the results of two *and* operations. When the intersection of the *and* operations is not empty then a duplication of data records results. Searching the report form and deleting duplicate reports would provide a 'patch' but would add more time to the retrieval process.

When designing the data query module, a major goal was not to restrict the user to a limited number of parameters for the search. This decision complicated the algorithm of implementing a query process with multiple many *ors* and *ands*. HyperCard was not designed for complicated query retrievals and the implementation became very awkward. In retrospect, there are two other design options that I would choose over the current model: limit the query choices available to the user, or use an external source code and develop a series of tables and indices.

Restricting the complication of the query operations will decrease the flexibility of the system, but will remove ambiguous procedures from the system. Using external code would maintain the flexibility and is probably the better option. When a query is requested, the table would be search instead of the database. This would speed up the retrieval process and allow for more complicated query request.

3. Quiz Design

If the acoustic quiz had been designed as a true - false or multiple choice quiz, then providing automatic grading would have been a simple matter. Due to the nature of acoustic analysis, a "fill in the blank" style test was a better learning tool and would be useful to a wider expertize level. It is

41

important for the analyst to be able to pick out the important lines without being cued. The multiple test would provide too many clues for actual analysis to be performed. Therefore the quiz was created in a free form style and grading must be done by manually.

C. THE UNCLASSIFIED DATABASE

To allow wider access of the initial system, these modules have been implemented as unclassified. In order to do this, a fictitious database was created. The database consists of seven types of whales and six sound sources by which the whale can be detected. Each of these "created" sound sources were then assigned a numerical value. Some of these source values are assigned to more than one whale type and others are unique to one type. For example, in this database, a Humpback whale's flipper produces a sound source at 3.5 and its fluke sound source is at 4.0, while a Sperm whale's flipper and fluke produce a source at 3.5 and 5.0 respectively. To further insure the classification of the database as unclassified, no units are assigned to the sound source numbers. See Appendix A, Figure 1 for a breakdown of the whale types and their assigned source values.

Next a detection database was created which gives details of whales detected from 1985 to 1988. This information is contained in a formatted message form which contains the whale name, type, id, source, source value (called cycles), dates of detection, and location of detection. The whale name was assumed to be assigned at the time of detection. For example, whale 3-86 is the name of the third whale detected in 1986. The id is a unique identifier that is assigned to an individual whale. The id can be used to retrieve data about multiple detections of the same whale over time.

42

Acoustic gram data was created by using actual grams with any possible classified data removed, thus creating a sanitized gram. An arbitrary number scale from 1 to 7 was inserted on top of the gram and whale signatures were drawn into each gram to correspond to one of the detection description in our fantasy database. The grams were then linked to the text message that they represent. The whale signatures were drawn by using the HyperCard paint function. See Appendix A for a break-down of the data used.

D. SUMMARY

In the ASW community, message data alone is not enough to give a good description of the data and gram 'pictures' are essential. This makes the acoustic database an ideal candidate for modeling to the concept of a multimedia database. Obviously this unclassified, 'creative' data is of no use to the ASW community, but it does provide the concept of linking acoustic, message and pictorial media together, and provides a user-interface for evaluation by the acoustic analysts. The modules have been designed so that classified data could easily replace the 'creative' data.

The design of the modules functionality does not limit its use to a ship. Any ASW platform, including shore sites could use the modules as designed. Although the demonstration modules only shows passive acoustic displays, active displays could be easily incorporated into the module.

VI. FUTURE DIRECTION/RESEARCH

A. INTRODUCTION

There is an effort underway in the Navy to stop the development of self contained, unique computer systems which exist today and instead develop highly integrated, compatible systems. This integration would provide communication and commonality between systems thus producing a more powerful, productive capability within the Navy. The designation of Ada as the official DOD language is a step towards this effort. Unfortunately, to develop and design a fully integrated computer system is quite complicated and extremely expensive. Concerns about the National deficit coupled with the easing of tensions with the Soviet Union, has created an overall reluctance in Congress to allocate the money to create such a system. It is important therefore to continue cost-effective research such as the AGROS project to provide the Navy with more updated systems, albeit not integrated. This chapter discusses the acoustic database as a part of a future integrated ASW system and its possible expansion through use of artificial intelligence, in the hopes that the Navy will one day achieve integration of its computer systems. The installation of the acoustic database as it exist today will be discussed in the next chapter.

B. THE FULLY INTEGRATED SYSTEM

As a part of an integrated system, the acoustic database would most likely reside in a separate processor such as a Suns-like workstation or a specialized database computer. It would be required to be compatible with both the signal processor and the communication processor so that the database could directly

access and retrieve gram data and message data. The compatibility between the database processor and the signal processor would allow additional processing information about the grams to be stored as part of the database. This data, such as bandwidth and signal-to-noise ratio would increase the sophistication of the database by providing additional information which will aid in the classification process.

In this integrated system, when a gram with a possible submarine signature is being evaluated, a copy of the real time gram data would be transferred to the database computer by the user. The operator would request a retrieval of similar items from the database and perform a visual side by side comparison. The additional processing information would help in the final decision making of the signature classification.

Adding items to the database would be a simple matter. The desired textual information could be taken directly from the communication system already in the required format and the gram data could just as easily be accessed from the signal processor.

C. EXPANDED ROLE

By merging database research efforts with artificial intelligence efforts, the acoustic database could be expanded beyond a simple database to produce an automated analyzing and alerting system.

In this expanded role, the database would additionally contain a collection of rules and facts for analyzing a submarine. The alerting system would search gram data from the signal processor looking for threat frequencies and using its facts, rules and gram data to analyze the line.

45

It would also search the database to locate a past detection that matches the current signature. When the alerting system reaches a specified confidence level, it will inform its detection to the analysts. The final determination of submarine or non-submarine would be made by the human expert.

The system would improve over time by "learning" new facts about gram analyses. If the signature is indeed a submarine, the analyst would confirm that to the computer and then include it in the database. If the signature was not a submarine, the factors which led the human analyst to that conclusion would be given to the computer. This information would be added to other rules thus allowing the computer to learn more about acoustics and become more proficient.

The database environment will provide the large amount of storage needed for rules and and facts and support the requirement to rapidly retrieve this information during analysis. The use of a specialized database machine for the storage and retrieval of facts and rules is described in detail in a paper by Hsiao and McGhee. [Ref. 7]

D. SUMMARY

The capabilities of the acoustic database as part of an integrated system could range from a simple system such as the current ARGOS acoustic module, or expanded to a more advanced artificial intelligence system, or any level in between. The realization of a fully integrated Navy will require a major software engineering feat. The determination of the role of an acoustic database and its implementation must be accomplished as part of a coordinated effort involving military systems experts and civilian contractors. The potential for such an integrated system is exciting dream for the future.

VII. CONCLUSIONS

The purpose of the acoustic modules was to demonstrate the advantages of a multimedia database of historical gram data. It was also meant to serve as a prototype system in support of the design of a more sophisticated system. The main concern of this module has been the user - machine interface. I believe that once actual submarine data is entered into the database, these modules could provide a useful analyzing tool which would improve the recognition differential of the analyst. The acoustic modules would contribute to the paper reduction effort by eliminating the need for reference and training material in the current paper form. Further, the ability to rapidly retrieve data would result in faster classification and reporting of submarines thus increasing the efficiency of the ASW community. Increased readiness will also be realized through improved training tools provided by the training module.

Although the acoustic modules as designed could be a useful tool, these modules were not meant to be used as a fully implemented program. There are many weaknesses that need to be addressed before these modules will achieve their full potential as a part of the ARGOS project. These issues will be addressed in the following section.

If the module is implemented, a centralized agency, such as NISC, would need to be responsible for the update of the acoustic database. Just as publications are periodically updated and released, the databases would be revised to include the latest data. It is anticipated that most commands would want to be able to maintain their own database in addition to the official database. This local database could contain signatures that represent certain peculiarities of that

command's sensor systems. To accomplish the ability to create a local database, there needs to be an efficient way to enter new gram data into the database. The current database was entered with the use of a scanner. It is unlikely that many of the operational commands have scanners available to them. An alternative to the scanner would be to develop a program that would convert the gram data from the signal processors to a format that the acoustic database could understand.

Timeliness of data retrieval is another major issue. In a real time submarine prosecution, time is of the utmost importance and any time delay will appear amplified during a stressful situation. Currently data is searched using the Hypertalk FIND function on the summary stack. Once a value is found, the gram data from a separate stack must be retrieved. The switching between stacks tends to slow down the retrieval process. To speed the retrieval process, an indexing scheme should be implemented possibly using an XCMD written in a more optimal coding language such as C.

Factors which led to the use of HyperCard vice SuperCard as the developmental environment for the acoustic modules included the increased memory needed to run SuperCard, the faster operating speed of HyperCard and the probability that Navy systems will not have color options available to them. However, SuperCard has many advantages such as multiple screen displays and color options that would be interesting to test in the ARGOS environment.

The ARGOS project as developed thus far, clearly demonstrates the diversity of Navy functions that can be automated to improve productivity throughout the Service. There are still many areas which have not yet been addressed that would benefit from continuing research on the ARGOS project.

APPENDIX A

THE TEST DATABASE

The acoustic modules of the ARGOS project have been developed at the unclassified level to allow for the widest possible dissemination of the demonstration system. To accomplish the implementation of these modules, it was necessary to create an unclassified test database of acoustic detections. Towards this goal, a database consisting of eight kinds of whales have been described using seven different sources. The units indictate some sound frequency unit of the generating source. Figure 1 shows a breakdown of the whale types and their assigned sound source values. Figure 2 gives a breakdown of the number of each type of whales that were detected for a specified year. Data was created for the time period covering 1985 - 1988. Figure 3 shows the detection records for the entire database. All data in Appendix A is fictional and has no base in actual acoustic detections of whales or any other acoustical source.

whale sources	Blue	Finback	Humpback	Orca	Right	Sei	Sperm
Blow	4.0	3.0	5.0	4.5	2.7	3.5	5.5
Breach	1.8	1.8	1.8	1.5	1.5	1.5	1.5
Dorsal Fin	3.5	2.5	4.5	4.0	2.2	3.0	5.0
Flippers	Unknown	3.5	3.5	3.5	3.5	3.5	3.5
Fluke	5.0	4.0	4.0	4.0	4.0	5.0	5.0
Phonation	4.1	4.2	4.3	2.0	4.3	4.5	4.1
Speed	15	15	15	Unknown	Unknown	35	12

Figure 1 Whale Sound Sources

49

	1985	1986	1987	1988
BLUE	4	3	0	0
FINBACK	1	0	0	0
HUMPBACK	5	4	2	2
ORCA	3	0	3	1
RIGHT	0	1	0	0
SEI	0	1	0	0
SPERM	2	0	3	2

Figure 2   Number of Whale Detections Per Year

name	type	id	source	cys	date
Whale 1-85	Humpback	112	Blow:	5.0	01 Jan 85 - 03 Feb 85
Whale 2-85	Blue	221	DorsalFin:	3.5	12 Jan 85 - 13 Mar 85
Whale 3-85	Humpback	115	Phonation:	4.3	15 Feb 85 - 20 Feb 85
			Blow:	5.0	
Whale 4-85	Orca	336	Phonation:	2.0	28 Jan 85 - 03 Feb 85
			Fluke	4.0	
Whale 5-85	Orca	333	Phonation:	2.0	02 Feb 85 - 14 Feb 85
			Fluke	4.0	
Whale 6-85	Orca	335	Phonation:	2.0	03 Feb 85 - 14 Feb 85
Whale 7-85	Humpback	111	Fluke:	4.0	14 Feb 85 - 15 Mar 85
			Blow:	5.0	
Whale 8-85	Humpback	113	Fluke:	4.0	14 Feb 85 - 13 Mar 85
Whale 9-85	Humpback	114	Fluke:	4.0	15 Feb 85 - 17 Mar 85
Whale 10-85	Blue	224	DorsalFin:	3.5	03 Mar 85 - 07 Mar 85
Whale 11-85	Sperm	441	Breach:	1.5	17 Mar 85 - 21 Apr 85
Whale 12-85	Sperm	443	Breach:	1.5	21 Mar 85 - 03 May 85
Whale 13-85	Blue	223	DorsalFin:	3.5	28 Mar 85 - 02 Apr 85
Whale 14-85	Finback	551	DorsalFin:	2.5	03 Apr 85 - 06 May 85
			Blow:	3.0	
			Fluke:	4.0	
Whale 15-85	Blue	223	DorsalFin:	3.5	05 Apr 85 - 05 May 85

Figure 3   1985 Detections

50

name	type	id	source	cys	date
Whale 1-86	Blue	223	Breach: DorsalFin	1.8 3.5	28 Jan 86 - 02 Feb 86
Whale 2-86	Humpback	114	Breach: Flippers: Fluke:	1.8 3.5 4.0	03 Feb 86 - 05 Mar 86
Whale 3-86	Humpback	112	Breach: Blow:	1.8 5.0	05 Feb 86 - 10 Mar 86
Whale 4-86	Blue	223	Breach: DorsalFin: Blow: Phonation:	1.8 3.5 4.0 4.1	06 Feb 86 - 18 Feb 86
Whale 5-86	Humpback	113	Fluke:	4.0	10 Feb 86 - 28 Feb 86
Whale 6-86	Sei	662	DorsalFin: Blow: Phonation:	3.0 3.5 4.5	13 Feb 86 - 15 Feb 86
Whale 7-86	Humpback	115	Phonation: Blow:	4.3 5.0	21 Feb 86 - 01 Apr 86
Whale 8-86	Right	773	Blow:	2.7	03 Mar 86 - 19 Mar 86
Whale 9-86	Sei	662	DorsalFin: Flippers:	3.0 3.5	02 Apr 86 - 03 May 86
Whale 10-86	Blue		DorsalFin: Fluke:	3.5 5.0	06 Apr 86 - 11 Apr 86
Whale 11-86	Humpback		Breach: Flippers: Fluke:	1.8 3.5 4.0	06 Apr 86 - 19 Apr 86
Whale 12-86	Humpback	113	Fluke: Phonation: Blow:	4.0 4.3 5.0	10 Apr 86 - 21 Apr 86

Figure 4 1986 Detections

name	type	id	source	cys	date
Whale 1-87	Humpback	111	Breach: Fluke: Blow:	1.8 4.0 5.0	12 Feb 87 - 03 Mar 87
Whale 2-87	Humpback	115	Breach: Phonation: Blow:	1.8 4.3 5.0	13 Feb 87 - 07 Mar 87
Whale 3-87	Orca	333	Phonation: DorsalFin:	2.0 4.0	17 Feb 87 - 28 Feb 87
Whale 4-87	Sperm	442	Breach: Phonation: DorsalFin:	1.5 4.1 5.0	17 Feb 87 - 18 Feb 87
Whale 5-87	Orca	331	Phonation:	2.0	18 Feb 87 - 28 Feb 87
Whale 6-87	Orca	335	Phonation:	2.0	18 Feb 87 - 27 Feb 87
Whale 7-87	Sperm	443	Breach:	1.5	19 Feb 87 - 15 Mar 87
Whale 8-87	Sperm	441	Breach: Fluke: Blow:	1.5 5.0 5.5	21 Feb 87 - 15 Mar 87
Whale 9-87	Humpback	111	Breach: Fluke: Blow:	1.8 4.0 5.0	07 Apr 87 - 21 Apr 87
Whale 10-87	Humpback	115	Breach: Phonation: Blow:	1.8 4.3 5.0	10 Apr 87 - 19 Apr 87

Figure 5   1987 Detections

name	type	id	source	cys	date
Whale 1-88	Humpback	112	Bre . Phonation: Blow:	1.8 4.3 5.0	19 Jan 88 - 18 Feb 88
Whale 2-88	Orca	335	Phonation:	2.0	20 Jan 88 - 25 Feb 88
Whale 3-88	Humpback	113	Breach: Fluke: Phonation:	1.8 4.0 4.3	22 Jan 88 - 19 Feb 88
Whale 4-88	Sperm	441	Breach: Phonation:	1.5 4.1	22 Feb 88 - 13 Mar 88
Whale 5-88	Sperm	443	Breach: Phonation:	1.5 4.1	25 Feb 88 - 12 Mar 88

Figure 6   1988 Detections

52

ACOUSTIC MODULE SCRIPT

SCRIPTS FOR STACK: whale Main menu

```
** BKGND #1, BUTTON #1: exit *********************************
on mouseUp
  go whale main menu
end mouseUp
** CARD #1 ***********************************
on reportall
  put "RETRIEVING ENTIRE DATABASE.   PRESS THE SHIFT KEY TO CANCEL"
  lock screen
  go summaryw
  repeat with zz = 1 to the number of cards
    go card zz
    addtoreport
    if the shiftkey is down then
      put "TRANSACTION CANCELLED"
      play "boing"
      go whale main menu
      exit reportall
    end if
  end repeat
  go reportw
  unlock screen
end reportall
on AddToReport
  put false into newcard
    global var1,var2,var3,var4,var5,var6,var7
    repeat with x = (the number of lines in var1) down to 1
    delete line x of var1
  end repeat
  repeat with x = (the number of lines in var2) down to 1
    delete line x of var2
  end repeat
  repeat with x = (the number of lines in var3) down to 1
    delete line x of var3
  end repeat
  repeat with x = (the number of lines in var4) down to 1
    delete line x of var4
  end repeat
  repeat with x = (the number of lines in var5) down to 1
    delete line x of var5
  end repeat
  repeat with x = (the number of lines in var6) down to 1
    delete line x of var6
```

```
end repeat
repeat with x = (the number of lines in var7) down to 1
  delete line x of var7
end repeat
put  fld name into var1
put  fld type into var2
put fld id into var3
put fld date into var4
put the number of lines in fld source into k
repeat with j = 1 to k
  put word 1 of line j of fld source into line j of var5
  if word 1 of line j of fld source is "dorsal" then
    put word 2 of line j of fld source  after line j of var5
    put word 3 of line j of fld source into line j of var6
  else
    put word 2 of line j of fld source into line j of var6
  end if
end repeat
put fld location into var7
 go last card of reportw
put 0 into max
put the number of lines in fld cycles into check
if check > max then put check into max
put the number of lines in fld source into check
if check > max then put check into max
put the number of lines in fld location into check
if check > max then put check into max
 put 0 into maxvar
put the number of lines in var5 into ck      -- field source
if ck > maxvar then put ck into maxvar
put the number of lines in var7 into ck       --field location
if ck > maxvar then put ck into maxvar
-- check if there is enough room to include this entry
if (max + maxvar) > 25 then
  doMenu "New Card"
   put 1 into max
   put true into newcard
else
   add 1 to max
end if
put var1 into line max of fld name
put var2 into line max of fld type
put var3 into line max of fld id
put var4 into line max of fld date
put var7 into line max of fld location
repeat with x = 1 to the number of lines in var5
  put line x of var5 into line max of fld source
  put line x of var6 into line max of fld cycles
  add 1 to max
end repeat
if newcard is true then
  go back
  put false into newcard
  go summaryw
end if
```

```
      go summaryw
      if the shiftkey is down then exit addtoreport
end addtoReport


** CARD #1, BUTTON #1: SEARCH ************************************
on mousedown
  put "SELECTED SEARCH" & return & "GET ENTIRE DATABASE" into Menu
  get HPopupMenu(Menu, 0,the mousev,the mouseh)
    if it is not zero then
    push card
    put item 1 of it into theline
    if theline = 1 then
      set the hilite of me to false
      go to card id 3652
    end if
    if theline  = 2 then
      set the hilite of me to false
      answer "THIS WILL TAKE AWHILE... ARE YOU SURE?" with "CONTINUE"¬
      or "CANCEL"
      if it is "CONTINUE" then reportall
    end if
  end if
end mousedown


** CARD #1, BUTTON #2: BROWSE ************************************
on mousedown
  put "GRAMS" & return & "SUMMARY MESSAGES" into Menu
  get HPopupMenu(Menu, 0,the mousev,the mouseh)
    if it is not zero then
    push card
    put item 1 of it into theline
    if theline = 1 then
      set the hilite of me to false
      go to stack "whale watch"
    end if
    if theline  = 2 then
      set the hilite of me to false
      go to stack "summaryw"
    end if
  end if
  end mousedown


** CARD #1, BUTTON #3: TRAINING ************************************
on mousedown
  global stu,instr
  put "STUDENT" & return & "INSTRUCTOR" into Menu
  get HPopupMenu(Menu, 0,the mousev,the mouseh)
  if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      set the hilite of me to false
      put True into stu
      put false into instr
      go train.grams
    end if
```

```
    if theline = 2 then
      set the hilite of me to false
      put False into stu
      put True into instr
      lock screen
      go train.grams
      ask password "Enter Password"
      if not (it is card field "Password") then
        go whale main menu
        answer "Invalid Password"
      end if
    end if
    unlock screen
  end if
end mousedown


** CARD #1, BUTTON #4: help ***********************************
on mouseUp
  push card
  go card id 8416
end mouseUp


** CARD #2 ***********************************
on HideOrShow FldName
  if visible of card field FldName is false then
    set visible of card field FldName to True
  else
    set visible of card field FldName to False
  end if
end HideOrShow


on TooManyChoices FldName
  answer "THIS FIELD CAN ONLY HAVE ONE SELECTION "
  answer "Replace old selection with new one?" with "YES" or "No"
  if it is "yes" then
    put the selection into line 1 of card field FldName
  else
  end if
end TooManyChoices



on EnterChoice FldName
  select the selectedline
  put (the number of lines of  card field FldName ) into i
  if (Fldname = "name1") and (i = 1) then
    TooManyChoices FldName
  else if (Fldname = "type1") and (i = 1) then
    TooManyChoices FldName
  else if (Fldname = "id1") and (i = 1) then
    TooManyChoices FldName
  else if (Fldname = "cycles1") and (i = 1) then
    TooManyChoices FldName
  else if (Fldname = "date1") and (i = 1) then
    TooManyChoices FldName
  else
```

56

```
      put the selection into line (i+1) of card field FldName
    end if
  end EnterChoice

  on PlaceOnScreen FldName, ButnName
    global pixelsNeeded
    if card fld lastone is empty then
      put FldName into line 1 of card fld LastOne
    else
      put the number of lines in card fld lastone into j
       repeat with x = 1 to j
        if line j of card fld LastOne = FldName then
        else
          put FldName into line (j + 1) of card fld LastOne
        end if
       end repeat
    end if
  end PlaceOnScreen

  ** CARD #2, FIELD #1: TYPE ************************************
  on mouseleave
    global menuname, ChoiceMade
    put the short name of me into menuname
    if the mouseloc is not within rect of card button menuname then
      hide me
    end if
    if ChoiceMade is true then
      PlaceOnScreen type1,"type:"
      show card fld type1
    end if
  end mouseleave

  on MouseEnter
    global ChoiceMade
    put false into ChoiceMade
  end MouseEnter

  on mousedown
    set locktext of me to false
    click at the clickloc
    select the selectedline
    set locktext of me to true
  end mousedown

  on mouseup
    global ChoiceMade
    EnterChoice type1
    put True into ChoiceMade
  end mouseup

  on mouseWithin
    if the selectedline is EMPTY then  select before text of me
    select the selectedline
  end mouseWithin
```

57

```
** CARD #2, FIELD #2: name **********************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
    hide me
  end if
  if ChoiceMade is true then
    PlaceOnScreen name1,"name:"
    show card fld NAME1
  end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
  set locktext of me to false
  click at the clickloc
  select the selectedline
  set locktext of me to true
end mousedown

on mouseup
  global ChoiceMade
  EnterChoice name1
  put True into ChoiceMade
end mouseup

on mouseWithin
  if the selectedline is EMPTY then  select before text of me
  select the selectedline
end mouseWithin

** CARD #2, FIELD #3 **********************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
    hide me
  end if
  if ChoiceMade is true then
    PlaceOnScreen id1,"id:"
    show card fld id1
  end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
```

```
    set locktext of me to false
    click at the clickloc
    select the selectedline
    set locktext of me to true
  end mousedown

  on mouseup
    global ChoiceMade
    EnterChoice id1
    put True into ChoiceMade
  end mouseup

  on mouseWithin
    if the selectedline is EMPTY then  select before text of me
    select the selectedline
  end mouseWithin

** CARD #2, FIELD #4: source *********************************
  on mouseleave
    global menuname, ChoiceMade
    put the short name of me into menuname
    if the mouseloc is not within rect of card button menuname then
      hide me
    end if
    if ChoiceMade is true then
      PlaceOnScreen source1,"source:"
      show card fld source1
    end if
  end mouseleave

  on MouseEnter
    global ChoiceMade
    put false into ChoiceMade
  end MouseEnter

  on mousedown
    set locktext of me to false
    click at the clickloc
    select the selectedline
    set locktext of me to true
  end mousedown

  on mouseup
    global ChoiceMade
    EnterChoice source1
    put True into ChoiceMade
  end mouseup

  on mouseWithin
    if the selectedline is EMPTY then  select before text of me
    select the selectedline
  end mouseWithin

** CARD #2, FIELD #5: Cycles *********************************
```

59

```
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
    hide me
  end if
  if ChoiceMade is true then
    show card fld cycles1
  end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
  set locktext of me to false
  click at the clickloc
  select the selectedline
  set locktext of me to true
end mousedown

on mouseup
  global ChoiceMade
  EnterChoice cycles1
  put True into ChoiceMade
end mouseup

on mouseWithin
  if the selectedline is EMPTY then  select before text of me
  select the selectedline
end mouseWithin

** CARD #2, FIELD #6: location **********************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
    hide me
  end if
  if ChoiceMade is true then
    PlaceOnScreen location1,"location:"
    show card fld location1
  end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
  set locktext of me to false
  click at the clickloc
```

60

```
  select the selectedline
  set locktext of me to true
end mousedown

on mouseup
  global ChoiceMade
  EnterChoice location1
  put True into ChoiceMade
end mouseup

on mouseWithin
  if the selectedline is EMPTY then  select before text of me
  select the selectedline
end mouseWithin

** CARD #2, FIELD #7: Date **********************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
    hide me
  end if
  if ChoiceMade is true then
    show card fld date1
  end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
  set locktext of me to false
  click at the clickloc
  select the selectedline
  set locktext of me to true
end mousedown

on mouseup
  global ChoiceMade
  EnterChoice date1
  put True into ChoiceMade
end mouseup

on mouseWithin
  if the selectedline is EMPTY then  select before text of me
  select the selectedline
end mouseWithin
** CARD #2, FIELD #8: Other **********************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
    hide me
```

61

```
    end if
    if ChoiceMade is true then
      PlaceOnScreen other1,"find word:"
      show card button "find word:"
      show card fld other1
    end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
  set locktext of me to false
  click at the clickloc
  select the selectedline
  set locktext of me to true
end mousedown

on mouseup
  global ChoiceMade
  EnterChoice other1
  adjustsize other1
    put True into ChoiceMade
end mouseup

on mouseWithin
  if the selectedline is EMPTY then  select before text of me
  select the selectedline
end mouseWithin

** CARD #2, BUTTON #1: NAME ************************************
on mouseEnter
  global MenuName
  put short name of me into MenuName
  set top of card fld MenuName to bottom of me
  set left of card field MenuName to left of me
  show card fld menuName
end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave

** CARD #2, BUTTON #2: Type ************************************
on mouseEnter
  global MenuName
  put short name of me into MenuName
  set top of card fld MenuName to bottom of me
  set left of card field MenuName to left of me
  show card fld menuName
```

```
  end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave

** CARD #2, BUTTON #3 **********************************
on mouseEnter
  global MenuName
  put short name of me into MenuName
  set top of card fld MenuName to bottom of me
  set left of card field MenuName to left of me
  show card fld menuName
end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave


** CARD #2, BUTTON #4: Date **********************************
on mouseEnter
  global MenuName
  put short name of me into MenuName
  set top of card fld MenuName to bottom of me
  set left of card field MenuName to left of me
  show card fld menuName
end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave

** CARD #2, BUTTON #5: Cycles **********************************
on mouseEnter
  global MenuName
  put short name of me into MenuName
  set top of card fld MenuName to bottom of me
  set left of card field MenuName to left of me
  show card fld menuName
end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
```

```
   end if
end MouseLeave

** CARD #2, BUTTON #6: Source *************************************
on mouseEnter
  global MenuName
  put short name of me into MenuName
  set top of card fld MenuName to bottom of me
  set left of card field MenuName to left of me
  show card fld menuName
end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave

** CARD #2, BUTTON #7: Location *************************************
on mouseEnter
  global MenuName
  put short name of me into MenuName
  set top of card fld MenuName to bottom of me
  set left of card field MenuName to left of me
  show card fld menuName
end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave

** CARD #2, BUTTON #8: Other *************************************
on mouseEnter
  global MenuName
  put short name of me into MenuName
  set top of card fld MenuName to bottom of me
  set left of card field MenuName to left of me
  show card fld menuName
end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave

** CARD #2, BUTTON #9: clear *************************************
on mouseUp
  lock screen
  hide card fld name1
```

```
     repeat with i = the number of lines in card fld name1 down to 1
       delete line i of card fld name1
     end repeat
     hide card fld type1
     repeat with i = the number of lines in card fld type1 down to 1
       delete line i of card fld type1
     end repeat
     hide card fld id1
     repeat with i = the number of lines in card fld id1 down to 1
       delete line i of card fld id1
     end repeat
     hide card fld date1
     repeat with i = the number of lines in card fld date1 down to 1
       delete line i of card fld date1
     end repeat
     hide card fld cycles1
     repeat with i = the number of lines in card fld cycles1 down to 1
       delete line i of card fld cycles1
     end repeat
     hide card fld source1
     repeat with i = the number of lines in card fld source1 down to 1
       delete line i of card fld source1
     end repeat
     hide card fld location1
     repeat with i = the number of lines in card fld location1 down to 1
       delete line i of card fld location1
     end repeat
     hide card fld other1
     repeat with i = the number of lines in card fld other1 down to 1
       delete line i of card fld other1
     end repeat
     hide card fld lastone
     repeat with i = the number of lines in card fld lastone down to 1
       delete line i of card fld lastone
     end repeat
     unlock screen
   end mouseUp


** CARD #2, BUTTON #10: show - hide source *********************************
on mouseUp
  HideOrShow source1
end mouseUp


** CARD #2, BUTTON #11: show - hide name1 *********************************
on mouseUp
  HideOrShow name1
end mouseUp


** CARD #2, BUTTON #18: help *********************************
on mouseUp
go card id 4657 -- help card
 end mouseUp


** CARD #2, BUTTON #20: clr data *********************************
on mouseUp
```

```
  global choices,FldName,numcount
  repeat with i = the number of lines in choices down to 1
    delete line i of choices
  end repeat
  repeat with i = the number of lines in FldName down to 1
    delete line i of FldName
  end repeat
  repeat with i = the number of lines in numcount down to 1
    delete line i of numcount
  end repeat
end mouseUp


** CARD #2, BUTTON #22: execute ***********************************
on mouseUp
  global choices,FldName,numcount,alsoFirstTime, addflag,cycstuff,¬
  dates,dt,dt2
  repeat with i = the number of lines in choices down to 1
    delete line i of choices
  end repeat
  repeat with i = the number of lines in FldName down to 1
    delete line i of FldName
  end repeat
  repeat with i = the number of lines in numcount down to 1
    delete line i of numcount
  end repeat
  set the cursor to 4
  lock screen
  put true into AlsoFirstTime
  put true into addflag
  put card fld lastone into FldName
  put card fld cycles1 into cycstuff
  put card fld date1 into dates
  put the number of lines in card fld lastone into linNum
  repeat with x = 1 to linNum
    put line x of card fld lastone into temp
    put the number of lines in card fld temp into chnum
    put (the number of lines in numcount + 1) into jj
    put chnum into line jj of numcount
    repeat with j = 1 to chnum
      put (the number of lines in choices + 1) into ptr
      put line j of card fld temp into line ptr of choices
    end repeat
  end repeat
  go last card of reportw
  if (the number of lines in fld name > 0) and (addflag is true) then
    answer "Data from last search in stack" with "CLEAR" or ¬
    "ADD TO" or "Save AS..."
    if it is "clear" then
      clearstack
      go temp1
      clearstack
    else if it is "SAVE AS..." then
      save
      clearstack
    end if
```

66

```
end if
set the cursor to 4
put false into addflag
go summaryw
put line 1 of FldName into firstfld
put line 1 of choices into firstitem
put line 1 of numcount into numitems
find firstitem in fld firstfld
if the result is empty then
  put the id of this card into FirstOne
  put FirstOne into CurrentCard
 go next card
  put true into FirstTime
 repeat until the id of this card is FirstOne
   if FirstTime is true then go back
   put false into FirstTime
   put true into match
   put 2 into k
   if numitems > 1 then
     put 1 into xx
     put 1 into k
   else
     put 2 into xx
     put 2 into k
   end if
   repeat with x = xx to the number of lines in FldName
     if the shiftkey is down then exit mouseup
     put line x of FldName into FldVar
      put line x of Numcount into count
     repeat with j = 1 to count
       if the shiftkey is down then exit mouseup
       put line k of choices into searchpar
       add 1 to k
       find searchpar in  field FldVar
       if the result <> empty then
         beep
         -- no match in rest of datbase
         put "no match"
         wait 3 sec
         ⌐   false into match
         exit mouseup
       end if
       if the id of this card <> CurrentCard then
         go back
         put false into match
         exit repeat
       end if
     end repeat
     if match is false then exit repeat
   end repeat
     if match is true then
     AddTOReport AlsoFirstTime
     put false into AlsoFirstTime
     go Back
     DoMenu "copy card"
```

```
            push card
            go last card of templ
            DoMenu "Paste card"
            pop card
          end if
          go to next card
          find firstitem in fld firstfld
          put the id of this card into CurrentCard
        end repeat
        go first card of templ
        if match is true then
          DoMenu "delete card"
        end if
            go reportw
          end if
      put the number of lines in cycstuff into cycx
      if cycx > 0 and ((the number of lines in FldName) > 0) then
        go first card of templ
        put word 1 of cycstuff into min
        put word 3 of cycstuff into max
        repeat with x = 1 to the number of cards in templ
          put the number of items in fld cycles into jj
          put false into found
          repeat with ii = 1 to jj
            if item ii of fld cycles >= min and item ii of fld cycles ¬
            <= max then
              put true into found
              exit repeat
            end if
          end repeat
          if found is not true then
            domenu "delete card"
          else
            go next
          end if
        end repeat
      end if


      --********-****************DATE CHECK*********************
      put the number of lines in dates into dtx
      if dtx > 0 and ((the number of lines in FldName) > 0) then
        dateconvert dates
        put dt  into min
        put dt2  into max
        go last card of templ
        put the id of this card into lastcard
        go first card of templ
        repeat until the id of this card is lastcard
          put line 1 of fld date into dtcpr
          dateconvert dtcpr
          put false into found
          if dt >= min and dt <= max then
            put true into found
          end if
          if dt2 >= min and dt2 <= max then
```

68

```
            put true into found
          end if
            if found is not true then
            domenu "delete card"
          else
            go next
          end if
        end repeat
        put line 1 of fld date into dtcpr
        dateconvert dtcpr
        put false into found

        if dt >= min and dt <= max then
          put true into found
        end if

        if dt2 >= min and dt2 <= max then
          put true into found
        end if


        if found is not true then
          domenu "delete card"
        end if
        go reportw
        clearstack
        go temp1
        repeat with  x = 1 to the number of cards
          go card x
          addtoreport
        end repeat
      end if

      if cycx > 0 and  ((the number of lines in FldName) < 1) then
        go summaryw
        put dt into min
        put dt2 into max
        repeat with x = 1 to (the number of cards )
          go to card x of stack "summaryw"
          put word 1 of cycstuff into min
          put word 3 of cycstuff into max
          put the number of items in fld cycles into jj
          put false into found
          repeat with ii = 1 to jj
            if item ii of fld cycles >= min and item ii of fld cycles ¬
            <= max then
              put true into found
              exit repeat
            end if
          end repeat
          if found is  true then
            AddTOReport AlsoFirstTime
            put false into AlsoFirstTime
            go Back
            DoMenu "copy card"
```

69

```
          push card
          go last card of templ
          DoMenu "Paste card"
          pop card
        end if
      end repeat
      go reportw
    end if
    _*****************************date alone *****************
    put the number of lines in dates into dtx
    if dtx > 0 and  ((the number of lines in FldName) < 1) then
      dateconvert dates
      put "in  date alone"
      wait 30 ticks
      go summaryw
      put dt into min
      put dt2 into max
      repeat with x = 1 to (the number of cards )
        go to card x of stack "summaryw"
        put line 1 of fld date into dtcpr
        dateconvert dtcpr
        put false into found
        if dt >= min and dt  <= max then
          put true into found
        end if
        if dt2 >= min and dt2  <= max then
          put true into found
        end if
        if found is  true then
          AddTOReport AlsoFirstTime
          put false into AlsoFirstTime
          go Back
          DoMenu "copy card"
          push card
          go last card of templ
          DoMenu "Paste card"
          pop card
        end if
      end repeat
      go reportw
    end if
      unlock screen
    play "harpsichord" tempo 150 "e c "
end mouseUp

on AddToReport AlsoFirstTime
  put alsofirsttime into addflag
  put false into newcard
  global var1,var2,var3,var4,var5,var6,var7
  -- clear variables
  repeat with x = (the number of lines in var1) down to 1
    delete line x of var1
  end repeat
  repeat with x = (the number of lines in var2) down to 1
    delete line x of var2
```

70

```
end repeat
repeat with x = (the number of lines in var3) down to 1
  delete line x of var3
end repeat
repeat with x = (the number of lines in var4) down to 1
  delete line x of var4
end repeat
repeat with x = (the number of lines in var5) down to 1
  delete line x of var5
end repeat
repeat with x = (the number of lines in var6) down to 1
  delete line x of var6
end repeat
repeat with x = (the number of lines in var7) down to 1
  delete line x of var7
end repeat
-- put this cards flds into variables
put  fld name into var1
put  fld type into var2
put fld id into var3
put fld date into var4
put the number of lines in fld source into k
repeat with j = 1 to k
  put word 1 of line j of fld source into line j of var5
  if word 1 of line j of fld source is "dorsal" then
    put word 2 of line j of fld source  after line j of var5
    put word 3 of line j of fld source into line j of var6
  else
    put word 2 of line j of fld source into line j of var6
  end if
end repeat
put fld location into var7
go last card of reportw
put 0 into max
put the number of lines in fld cycles into check
if check > max then put check into max
put the number of lines in fld source into check
if check > max then put check into max
put the number of lines in fld location into check
if check > max then put check into max   -- get max lines in this card
put 0 into maxvar
put the number of lines in var5 into ck      -- field source
if ck > maxvar then put ck into maxvar
put the number of lines in var7 into ck       --field location
if ck > maxvar then put ck into maxvar
-- check if there is enough room to include this entry
if (max + maxvar) > 25 then
  doMenu "New Card"
  put 1 into max
  put true into newcard
else
  add 1 to max
end if

put var1 into line max of fld name
```

```
      put var2 into line max of fld type
      put var3 into line max of fld id
      put var4 into line max of fld date
      put var7 into line max of fld location
      repeat with x = 1 to the number of lines in var5
        put line x of var5 into line max of fld source
        put line x of var6 into line max of fld cycles
        add 1 to max
      end repeat
      if newcard is true then
        go back
        put false into newcard
      end if
    end addtoReport


    on clearvar var
      repeat with x = (the number of lines in var) down to 1
        delete line x of var
      end repeat
    end clearvar


    --************************DATECONVERT****************************
    on dateconvert dtinput
      global dt,dt2
      put dtinput
      wait 3 sec
      put word 2 of dtinput into mth
      if mth = "Jan" then put 1 into mth
      if mth = "feb" then put 2 into mth
      if mth = "mar" then put 3 into mth
      if mth = "apr" then put 4 into mth
      if mth = "may" then put 5 into mth
      if mth = "jun" then put 6 into mth
      if mth = "jul" then put 7 into mth
      if mth = "aug" then put 8 into mth
      if mth = "sep" then put 9 into mth
      if mth = "oct" then put 10 into mth
      if mth = "nov" then put 11 into mth
      if mth = "dec" then put 12 into mth
      put mth & "/" & word 1 of dtinput & "/" & word 3 of dtinput into dt
      convert dt to seconds
      put word 6 of dtinput into mth2
      if mth2 = "Jan" then put 1 into mth2
      if mth2 = "feb" then put 2 into mth2
      if mth2 = "mar" then put 3 into mth2
      if mth2 = "apr" then put 4 into mth2
      if mth2 = "may" then put 5 into mth2
      if mth2 = "jun" then put 6 into mth2
      if mth2 = "jul" then put 7 into mth2
      if mth2 = "aug" then put 8 into mth2
      if mth2 = "sep" then put 9 into mth2
      if mth2 = "oct" then put 10 into mth2
      if mth2 = "nov" then put 11 into mth2
      if mth2 = "dec" then put 12 into mth2
      put mth2 & "/" & word 5 of dtinput & "/" & word 7 of dtinput into dt2
```

```
convert dt2 to seconds
put "dt2 is " && dt2
wait 3 sec
end dateconvert
```

SCRIPTS FOR STACK: QuizReport

```
** BKGND #1, BUTTON #1: help ***********************************
on mouseUp
 push card
 go card id 8416 -- the help card
end mouseUp

** BKGND #1, BUTTON #2: quiz menu ***********************************
on mouseUp
 go to card id 3993 of stack "train.grams" -- quiz menu
end mouseUp

** BKGND #1, BUTTON #3: exit ***********************************
on mouseUp
 go whale main menu
end mouseUp

** CARD #1, FIELD #1: names ***********************************
on mousedown -- Finds the quiz results of the person highlighted by mouse
 set locktext of me to false
 click at the clickloc
 select the selectedline
 set locktext of me to true
end mousedown
on mouseup
 lock screen
 global quizname
 select the selectedline
 put word 2 of the selectedline into j
 go line j of card fld hide_fld
end mouseup

** CARD #1, BUTTON #2: clear ***********************************
on mouseUp
 repeat with i = the number of lines in card fld names down to 1
   delete line i of card fld names
   delete line i of card fld hide_fld
 end repeat
end mouseUp

** CARD #1, BUTTON #3: exit ***********************************
on mouseUp
 go whale main menu
end mouseUp

** CARD #1, BUTTON #4: help ***********************************
```

```
on mouseUp
  push card
  go card id 8416 -- help card
end mouseUp
**********
```

SCRIPTS FOR STACK: reportw

```
** STACK SCRIPT ***********************************
on save
  DoMenu "Save a Copy..."
end save

on clearstack
  repeat with x = (the number of cards) down to 2
    doMenu "Delete Card"
  end repeat
  doMenu "new card"
  go next card
  doMenu "delete card"
end clearstack

** BKGND #1, BUTTON #1: delete report*********************************
on mouseUp
  lock screen
  clearstack
  unlock screen
end mouseUp

** BKGND #1, BUTTON #2: to retrieved data *********************************
on mouseUp
  go temp1 -- this stack contains the gram and message data of the last data retreival
end mouseUp

** BKGND #1, BUTTON #3: menu *********************************
on mouseUp
  go to card id 2872 of  stack "whale main menu"
end mouseUp

***************
```

SCRIPTS FOR STACK: summaryw

```
** BKGND #1, BUTTON #1: Next *********************************
on mouseUp
  go to next card
end mouseUp

** BKGND #1, BUTTON #2: Prev *********************************
on mouseUp
  go to prev card
end mouseUp

** BKGND #1, BUTTON #3: MAP *********************************
```

```
on mouseUp
 push card
 go to card id line 3 of fld hid_data of stack "maps"
end mouseUp


** BKGND #1, BUTTON #4: GRAM ***********************************
on mouseUp
 push card
 go to card id line 1 of fld hid_data of stack "whale watch"
end mouseUp


** BKGND #1, BUTTON #5: pubs ***********************************
on mouseUp
 push card
 go to stack tim-3
end mouseUp


** BKGND #1, BUTTON #6: picture ***********************************
on mouseUp
 push card
 go to card id line 2 of fld hid_data of stack "pictures"
end mouseUp


** BKGND #1, BUTTON #7: return ***********************************
on mouseUp
 go back
end mouseUp


** BKGND #1, BUTTON #8: New Button ***********************************
on mouseUp
 go to card id 2872 of stack "whale main menu"
end mouseUp


** BKGND #1, BUTTON #9: help ***********************************
on mouseUp
go card 1456 -- help card
 end mouseUp


** BKGND #1, BUTTON #10: exit ***********************************
on mouseUp
 go home
end mouseUp


** BKGND #1, BUTTON #11: report ***********************************
on mouseUp
 go reportw
end mouseUp
 ****************
```

SCRIPTS FOR STACK: pictures

```
** BKGND #1, BUTTON #1: Return ***********************************
on mouseUp
 pop card
end mouseUp
```

75

****

****
SCRIPTS FOR STACK: tracks

** BKGND #1, BUTTON #1: return **************************************
```
on mouseUp
 pop card
end mouseUp
```

**************✦****
SCRIPTS FOR STACK: TRAIN.GRAMS

** STACK SCRIPT *************************************
```
on clearData
 lock screen
 repeat with i = the number of lines in card fld testgrams down to 1
   delete line i of card fld testgrams
 end repeat
 repeat with i = the number of lines in card fld questions down to 1
   delete line i of card fld questions
 end repeat
 unlock screen
end clearData
```

** BKGND #1, BUTTON #1: help *************************************
```
on mouseUp
go card id 3456 -- help card
 end mouseUp
```

** BKGND #1, BUTTON #2: exit *************************************
```
on mouseUp
 go whale main menu
end mouseUp
```

** BKGND #1, BUTTON #3: go train main menu *************************************
```
on mouseUp
 if the id of me is 3993
 then answer "Already a training main menu"
 else
   go to card id 3993
 end if
end mouseUp
```

** CARD #1 *************************************

```
on OpenCard
 global instr,stu
 lock screen
 if stu is true then
   hide card button "Create"
   hide card button "Preview"
   hide card button "Delete"
```

```
      hide card button "Results"
      show card button "Take Quiz"
      show card button "Tutorial"
      put "STUDENT MODE" into card fld "mode"
    else if instr is true then
      show card button "Create"
      show card button "Preview"
      show card button "Delete"
      show card button "Results"
      hide card button "Take Quiz"
      hide card button "Tutorial"
      put "INSTRUCTOR MODE" into card fld "mode"
    end if
    unlock screen
end OpenCard

on ChPassWd
  ask password "Enter New Password"
  put it into card field "Password"
end ChPasswd


** CARD #1, FIELD #2: Take quiz **~~**********************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
    hide me
  end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
  set locktext of me to false
  click at the clickloc
  select the selectedline
  set locktext of me to true
end mousedown
on mouseup
  lock screen
  global quizname,testvar,grams,quest,count,questnum,results,nextgram
  global name,NewCardid,scriptvar
  select the selectedline
  put the selection into quizname

  --              **************
  put false into found
  ask "ENTER YOUR NAME"
  if the selection is empty then exit mouseup
  put it into name
  lock screen
```

```
push card
go quizreport
go last
DoMenu "new card"
put "Quiz"&&quizname&&" results for"&&name into fld header
put the id of this card into NewCardid
go first
put the number of lines in card fld names into j
put "Quiz"&&quizname&&" results for"&&name into line j+1 of card fld¬
names
put NewCardid into line j+1 of card fld hide_fld
pop card
put empty into scriptvar
repeat with i = 1 to the number of lines in card fld "testTable"
  put item 1 of line i of card fld "testTable" into testvar
  if quizname is in testvar then
    put true into found
    exit repeat
  end if
end repeat
if found is true then
  put 1 into count
  go item 2 of line i of card fld "testTable"
  put card field "testgrams" into grams
  put card field "questions" into quest
  put item 3 of line count of grams into nextgram
  push card
  go "whale watch"
  put 1 into questnum
  go card id nextgram
  repeat with i = the number of lines in fld "answer" down to 1
    delete line i of fld "answer"
  end repeat
  hide bg button "return"
  hide bg button "picture"
  hide bg button "chart"
  hide bg button "summary"
  hide bg button "prev"
  hide bg button "next"
  hide bg button "help"
  hide bg button "pubs"
  hide bg button "exit"
  repeat with i = the number of lines in fld answer down to 1
    delete line i of fld answer
  end repeat
  put quest into fld question
  hide fld question
  show bg button "end quiz"
  show bg button "show questions"
  show bg button "next gram"
  show fld "answer"
  show bg button "show status"
  show fld "num"
  put 1 into count
  put item 3 of line count of grams into nextgram
```

```
      put "Gram" && count & ":" & return after fld Quizscript
      add 1 to count
    else
      answer "Test Named" && """"&quizname&"""" && "Not Found"
    end if
    unlock screen
  end mouseUp


  on mouseWithin
    if the selectedline is EMPTY then  select before text of me
    select the selectedline
  end mouseWithin

  ** CARD #1, FIELD #3: Delete ************************************
  on mouseWithin
    if the selectedline is EMPTY then  select before text of me
    select the selectedline
  end mouseWithin

  on mouseleave
    global menuname, ChoiceMade
    put the short name of me into menuname
    if the mouseloc is not within rect of card button menuname then
      hide me
    end if
  end mouseleave

  on MouseEnter
    global ChoiceMade
    put false into ChoiceMade
  end MouseEnter

  on mousedown
    set locktext of me to false
    click at the clickloc
    select the selectedline
    set locktext of me to true
  end mousedown

  on mouseUp
    global quizname,testTable,testvar
    lock screen
    select the selectedline
    put the selection into quizname
    put false into found

    if the selection is empty then exit mouseup
    put "SEARCHING" into card fld feedback
    show card fld feedback
    repeat with i = 1 to the number of lines in card fld "testTable"
      put item 1 of line i of card fld "testTable" into testvar
      if quizname is in testvar then
        put true into found
        exit repeat
      end if
```

79

```
    end repeat
    if found is true then
      put item 2 of line i of card fld "testTable" into check
      go check
      answer "ARE YOU SURE YOU WANT TO DELETE" && quizname&"?" ¬
      with DELETE or CANCEL
      if it is "DELETE" then
        put check
        wait 3 sec
        put the short name of this card into thiscard
        put thiscard
        wait 3 sec
        if word 3 of check is word 3 of thiscard then
          doMenu "cut card"
          put "it checks out"
        end if
        go card id 3993
        delete line i of card fld "testTable"
        delete line i of card fld "take quiz"
        delete line i of card fld "delete"
        delete line i of card fld "Preview"
        put quizname && "HAS BEEN DELETED" into card fld feedback
        wait 1 sec
      hide card fld feedback
      end if
    else
      answer "Test Named" && """"&quizname&"""" && "Not Found"
    end if
    end mouseUp


** CARD #1, FIELD #6: Preview *************************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
    hide me
  end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
  set locktext of me to false
  click at the clickloc
  select the selectedline
  set locktext of me to true
end mousedown
on mouseup
  lock screen
  global quizname,testvar,grams,quest,count,questnum,results,nextgram
  global name,NewCardid,scriptvar,pre_mode
  put true into pre_mode
```

```
select the selectedline
put the selection into quizname
put false into found
lock screen
put empty into scriptvar
repeat with i = 1 to the number of lines in card fld "testTable"
  put item 1 of line i of card fld "testTable" into testvar
  if quizname is in testvar then
    put true into found
    exit repeat
  end if
end repeat
if found is true then
  put 1 into count
  go item 2 of line i of card fld "testTable"
  put card field "testgrams" into grams
  put card field "questions" into quest
  put item 3 of line count of grams into nextgram
  put item 3 of line count of grams
  wait 3 sec
  push card
  go "whale watch"
  put 1 into questnum
  wait 3 sec
  go card id nextgram
  repeat with i = the number of lines in fld "answer" down to 1
    delete line i of fld "answer"
  end repeat
  hide bg button "return"
  hide bg button "picture"
  hide bg button "chart"
  hide bg button "summary"
  hide bg button "prev"
  hide bg button "next"
  hide bg button "help"
  hide bg button "pubs"
  hide bg button "exit"
  repeat with i = the number of lines in fld answer down to 1
    delete line i of fld answer
  end repeat
  put quest into fld question
  hide fld question
  show bg button "end quiz"
  show bg button "show questions"
  show bg button "next gram"
  show fld "answer"
  show bg button "show status"
  show fld "num"
  put 1 into count
  put item 3 of line count of grams into nextgram
  put "Gram" && count & ":" & return after fld Quizscript
  add 1 to count
else
  answer "Test Named" && ""&quizname&""&& "Not Found"
end if
```

```
    unlock screen
  end mouseUp


  on mouseWithin
    if the selectedline is EMPTY then  select before text of me
    select the selectedline
  end mouseWithin

** CARD #1, FIELD #7: delete tutorial **********************************
  on mouseWithin
    if the selectedline is EMPTY then  select before text of me
    select the selectedline
  end mouseWithin
  on mouseleave
    global menuname, ChoiceMade
    put "delete" into menuname
    if the mouseloc is not within rect of card button menuname then
      hide me
    end if
  end mouseleave


  on MouseEnter
    global ChoiceMade
    put false into ChoiceMade
  end MouseEnter


  on mousedown
    set locktext of me to false
    click at the clickloc
    select the selectedline
    set locktext of me to true
  end mousedown


  on mouseUp
    global quizname,TeachTable,testvar
    lock screen
    select the selectedline
    put the selection into quizname
    put false into found
     if the selection is empty then exit mouseup
     repeat with i = 1 to the number of lines in card fld "teachTable"
      put item 1 of line i of card fld "teachTable"  into testvar
      if quizname is in testvar  then
        put true into found
        exit repeat
      end if
    end repeat
    if found is true then
      put item 2 of line i of card fld "testTable" into check
      go item 2 of line i of card fld "testTable"
      answer "ARE YOU SURE YOU WANT TO DELETE" && quizname&"?" ¬
      with DELETE or CANCEL
      if it is "DELETE" then
        put the short name of this card into thiscard
        if check is thiscard
```

82

```
        then doMenu "cut card"
        go card id 3993
        delete line i of card fld "teachTable"
        delete line i of card fld "tutorial"
        delete line i of card fld "delete tutorial"
        delete line i of card fld "Preview tutorial"
        put quizname && "HAS BEEN DELETED" into Menu
        get HPopupMenu(Menu, 0,120,158)
        end if
     else
       answer "Test Named" && """&quizname&""" && "Not Found"
     end if
  end mouseUp

** CARD #1, FIELD #10: tutorial ***********************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
     hide me
  end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
  set locktext of me to false
  click at the clickloc
  select the selectedline
  set locktext of me to true
end mousedown
on mouseup
  lock screen
  global quizname,testvar,grams,quest,count,questnum,results,nextgram
  global name,NewCardid,scriptvar,help_mode
  select the selectedline
  put the selection into quizname
  put true into help_mode
  put false into found
  put empty into scriptvar
  repeat with i = 1 to the number of lines in card fld "testTable"
     put item 1 of line i of card fld "testTable" into testvar
     if quizname is in testvar then
        put true into found
        exit repeat
     end if
  end repeat
  if found is true then
     put 1 into count
     go item 2 of line i of card fld "testTable"
     put card field "testgrams" into grams
     put card field "questions" into quest
```

83

```
        put item 3 of line count of grams into nextgram
        push card
        go "whale watch"
        put 1 into questnum
        go card id nextgram
        repeat with i = the number of lines in fld "answer" down to 1
          delete line i of fld "answer"
        end repeat
        hide bg button "return"
        hide bg button "picture"
        hide bg button "chart"
        hide bg button "summary"
        hide bg button "prev"
        hide bg button "next"
        hide bg button "menu"
        hide bg button "help"
        hide bg button "pubs"
        hide bg button "exit"
        repeat with i = the number of lines in fld answer down to 1
          delete line i of fld answer
        end repeat
        put quest into fld question
        hide fld question
        show bg button "end quiz"
        show bg button "show questions"
        show bg button "next gram"
        show fld "answer"
        show bg button "show status"
        show fld "num"
        put 1 into count
        put item 3 of line count of grams into nextgram
        put "Gram" && count & ":" & return after fld Quizscript
        add 1 to count
      else
        answer "Test Named" && ""&quizname&"" && "Not Found"
      end if
      unlock screen
    end mouseUp

    on mouseWithin
      if the selectedline is EMPTY then  select before text of me
      select the selectedline
    end mouseWithin
```

```
    on mouseEnter
      global MenuName
      put short name of me into MenuName
      set top of card fld MenuName to bottom of me
      set left of card field MenuName to left of me
      show card fld menuName
    end mouseEnter

    on MouseLeave
      global MenuName
      if the mouseloc is not within rect of card fld MenuName then
```

```
      hide card field MenuName
    end if
  end MouseLeave


** CARD #1, BUTTON #2: Create *****************************^*********
on mousedown
  global quizname,add_help
  put "CREATE QUIZ" & return & "CREATE TUTORIAL" into Menu
  get HPopupMenu(Menu, 0,the mousev,the mouseh)
    if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      set the hilite of me to false
      put false into add_help
    end if
    if theline  = 2 then
      set the hilite of me to false
      put true into add_help
    end if
  end if
    if add_help is true then
    ask "ENTER NEW TUTORIAL NAME:"
  else
    ask "ENTER NEW QUIZ NAME:"
  end if
  if it is empty then exit mousedown
  put it into quizname
  lock screen
  go  card id 3159
  doMenu "copy card"
  go last card
  doMenu "paste card"
  put the id of this card into NewCardID
  go card id 3993
  if add_help is false then
    put the number of lines in card fld testTable into i
    put quizname && "," && NewCardID into line (i+1) of ¬
    card fld testTable
    put quizname into line (i+1) of card fld "take quiz"
    put quizname into line (i+1) of card fld "delete"
    put quizname into line (i+1) of card fld "preview"
  else
    put the number of lines in card fld teachTable into i
    put quizname && "," && NewCardID into line (i+1) of ¬
    card fld TeachTable
    put quizname into line (i+1) of card fld "Tutorial"
    put quizname into line (i+1) of card fld "delete tutorial"
    put quizname into line (i+1) of card fld "preview tutorial"
  end if
  Go Back
  clearData
  set the bottom of card fld "thisname" to the top of card fld testgrams
  if add_help is false then
    put "GRAMS FOR QUIZ"&&quizname into card fld thisname
  else
```

85

```
    put "GRAMS FOR TUTORIAL"&&quizname into card fld thisname
   end if
   unlock screen
end mousedown


** CARD #1, BUTTON #3: Delete *************************************
 on mousedown
  global quizname,MenuName
  put "QUIZ" & return & "TUTORIAL" into Menu
  get HPopupMenu(Menu, 0,the mousev,the mouseh)
   if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      set the hilite of me to false
      put "delete" into MenuName
    end if
    if theline  = 2 then
      set the hilite of me to false
      put "delete tutorial" into MenuName
    end if
    set top of card fld MenuName to bottom of me
    set left of card field MenuName to left of me
    show card fld menuName
   end if
 end mousedown


on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave


** CARD #1, BUTTON #4: Results *************************************
on mouseUp
 go quizreport
end mouseUp


** CARD #1, BUTTON #5: Preview *************************************
on mousedown
  global quizname,MenuName
  put "QUIZ" & return & "TUTORIAL" into Menu
  get HPopupMenu(Menu, 0,the mousev,the mouseh)
   if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      set the hilite of me to false
      put "preview" into MenuName
    end if
    if theline  = 2 then
      set the hilite of me to false
      put "Preview tutorial" into MenuName
    end if
    set top of card fld MenuName to bottom of me
    set left of card field MenuName to left of me
```

86

```
      show card fld menuName
    end if
end mousedown

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave
** CARD #1, BUTTON #6: Tutorial *************************************
on mouseEnter
  global MenuName
  put short name of me into MenuName
  set top of card fld MenuName to bottom of me
  set left of card field MenuName to left of me
  show card fld menuName
end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave

** CARD #2 ***********************************
on EnterChoice FldName,name, GramID
  put (the number of lines of  card field FldName ) into i
    put "GRAM" && i+1 &", "&& name && " ," & GramID ¬
  into line (i+1) of card field FldName
end EnterChoice

on EnterChoice1 FldName,name
  put (the number of lines of  card field FldName ) into i
  put (i+1)&& name into line (i+1) of card field FldName
end EnterChoice1

** CARD #2, FIELD #1: SELECT TEST GRAMS *************************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
    hide me
  end if
end mouseleave

on MouseEnter
  global ChoiceMade
  put false into ChoiceMade
end MouseEnter

on mousedown
  set locktext of me to false
  click at the clickloc
```

```
   select the selectedline
   set locktext of me to true
 end mousedown

 on mouseup
  lock screen
  global ChoiceMade, GramID, scalevar
  select the selectedline
  put the selection into choicemade
  push card
   go to "summaryw"
  find choicemade in fld name
  put line 1 of fld hid_data into GramID
  pop card
  EnterChoice testgrams, ChoiceMade, GramID
  put True into ChoiceMade
  unlock screen
 end mouseup

 on mouseWithin
  if the selectedline is EMPTY then  select before text of me
  select the selectedline
 end mouseWithin


** CARD #2, FIELD #2: testgrams **************************************
on mousedown
  set locktext of me to false
  click at the clickloc
  select the selectedline
  set locktext of me to true
end mousedown

on mouseup
  lock screen
  global choicemade
  select the selectedline
  put item 3 of the selection into choicemade
  push card
   go "whale watch"
  go card id choicemade
  unlock screen
end mouseup


on mouseWithin
  if the selectedline is EMPTY then  select before text of me
  select the selectedline
end mouseWithin

** CARD #2, FIELD #5: TEST QUESTIONS **************************************
on mouseleave
  global menuname, ChoiceMade
  put the short name of me into menuname
  if the mouseloc is not within rect of card button menuname then
```

88

```
      hide me
    end if
  end mouseleave

  on MouseEnter
    global ChoiceMade
    put false into ChoiceMade
  end MouseEnter

  on mousedown
    set locktext of me to false
    click at the clickloc
    select the selectedline
    set locktext of me to true
  end mousedown

  on mouseup
    lock screen
    global ChoiceMade, GramID, scalevar
    select the selectedline
    put the selection into choicemade
    push card
    EnterChoice1 questions, ChoiceMade
    put True into ChoiceMade
    unlock screen
  end mouseup

  on mouseWithin
    if the selectedline is EMPTY then select before text of me
    select the selectedline
  end mouseWithin
```

** CARD #2, BUTTON #1: SELECT TEST GRAMS ***************************************

```
  on mouseEnter
    global MenuName
    put short name of me into MenuName
    set top of card fld MenuName to bottom of me
    set left of card field MenuName to left of me
    show card fld menuName
  end mouseEnter

  on MouseLeave
    global MenuName
    if the mouseloc is not within rect of card fld MenuName then
      hide card field MenuName
    end if
  end MouseLeave
```

** CARD #2, BUTTON #2: TEST QUESTIONS ***************************************
```
  on mouseEnter
    global MenuName
    put short name of me into MenuName
    set top of card fld MenuName to bottom of me
    set left of card field MenuName to left of me
```

```
    show card fld menuName
end mouseEnter

on MouseLeave
  global MenuName
  if the mouseloc is not within rect of card fld MenuName then
    hide card field MenuName
  end if
end MouseLeave

  put (the number of lines of  card field FldName ) into i
  -- put check here

  put (i+1)&& name into line (i+1) of card field FldName

end EnterChoice1



***************
```

SCRIPTS FOR STACK: Whale Watch

```
** STACK SCRIPT ***********************************
-- This stacks contains the acoustic grams
on closestack
  lock screen
  if visible of fld quizscript is true then
    set visible of fld quizscript to false
    set name of bg button "hide status" to "Show Status"
  end if
  if visible of fld question is true then
    set visible of fld question to false
    set name of bg button "hide questions" to "Show questions"
  end if
  show bg button "return"
  show bg button "picture"
  show bg button "chart"
  show bg button "summary"
  show bg button "prev"
  show bg button "next"
  show bg button "help"
  show bg button "pubs"
  show bg button "exit"
  hide fld "question"
  hide fld "answer"
  hide bg button "end quiz"
  hide bg button "show questions"
  hide bg button "show status"
  hide bg button "next gram"
  unlock screen
end closestack


** BACKGROUND #1 ***********************************
```

```
on openStack
  hide message box
  show menuBar
  pass openStack
end openStack


** BKGND #1, FIELD #2: answer **********************************
on enterinFielu
  global results, questnum, quest, answer
  put line (questnum - 1) \f quest & return after results
  put fld "answer" & return after results
end enterinField


** BKGND #1, FIELD #3: question **********************************
on mousedown
  set locktext of me to false
  click at the clickloc
  select the selectedline
  set locktext of me to true
end mousedown
on mouseup
  global CurrentQuestion, scriptvar
  select the selectedline
  put the selection into CurrentQuestion
  put "Question: "&& CurrentQuestion & return after fld quizscript
  put "Question: "&& CurrentQuestion & return after scriptvar
  ask "Enter Answer"
  if it is empty then
    put "ANSWER:  NOT ANSWERED"&return after fld quizscript
    put "ANSWER:  NOT ANSWERED"&return after scriptvar
  else
    put "ANSWER: "&& it & return after fld quizscript
    put "ANSWER: "&& it & return after scriptvar
  end if
  end mouseup


** BKGND #1, BUTTON #1: Next **********************************
on mouseUp
  go to next card
end mouseUp


** BKGND #1, BUTTON #2: Prev **********************************
on mouseUp
  go to prev card
end mouseUp


** BKGND #1, BUTTON #3: return **********************************
on mouseUp
  pop card
end mouseUp


** BKGND #1, BUTTON #4: chart **********************************
on mouseUp
  push card
  go to card id line 3 of fld hd_fld of stack "maps"
```

```
end mouseUp

** BKGND #1, BUTTON #5: Pubs **********************************
on mouseUp
 push card
 go to stack TIM-3
end mouseUp

** BKGND #1, BUTTON #6: summary **********************************
on mouseUp
 go to card id line 1 of fld hd_fld of stack "summaryw"
end mouseUp

** BKGND #1, BUTTON #7: picture **********************************
on mouseUp
 push card
 go to card id line 2 of fld hd_fld of stack "pictures"
end mouseUp

** BKGND #1, BUTTON #8: help **********************************
on mouseUp
 go card id 4453 -- help card
end mouseUp

** BKGND #1, BUTTON #9: Show MouseLoc **********************************
on mouseUp
 repeat until the mouse is down
  put the mouseLoc
 end repeat
end mouseUp

** BKGND #1, BUTTON #10: Show questions **********************************
on mouseUp
 global quest,questnum,results
  if  visible of fld "question" is false then
   set visible of fld "question"  to true
   set name of bg button "show questions" to "Hide questions"
  else
   set visible of fld "question"  to false
   set name of bg button "Hide questions" to "Show questions"
  end if
end mouseUp

** BKGND #1, BUTTON #11: End Quiz **********************************
on mouseUp
 global tester,grams,scriptvar,pre_mode
 answer "QUIT QUIZ?" with "YES" or "NO"
 if it is "no" then exit mouseup
 if pre_mode is false then
   lock screen
   push card
   go quizreport
   go last card
   put scriptvar into  fld testreport
   pop card
```

92

```
      end if
      repeat with j = 1 to (the number of lines in grams
        go card id item 3 of line j of grams
        put fld quizscript into tester
        repeat with i = (the number of lines in fld quizscript)¬
          down to 1
          delete line i of fld quizscript
        end repeat
        end repeat
      if visible of fld quizscript is true then
        set visible of fld quizscript to false
        set name of bg button "hide status" to "Show Status"
      end if
      if visible of fld question is true then
        set visible of fld question to false
        set name of bg button "hide Questions" to "Show Questions"
      end if
      show bg button "return"
      show bg button "picture"
      show bg button "chart"
      show bg button "summary"
      show bg button "prev"
      show bg button "next"
      show bg button "help"
      show bg button "pubs"
      show bg button "exit"
      hide fld "answer"
      hide me
      hide bg button "show questions"
      hide bg button "show status"
      hide bg button "next gram"
      hide field "num"
      put false into pre_mode   -- instructor preview mode
      go whale main menu
      unlock screen
    end mouseUp


** BKGND #1, BUTTON #12: Show Status **********************************
on mouseUp
  if visible of fld quizscript is false then
    set visible of fld quizscript to true
    set name of bg button "show status" to "Hide Status"
  else
    set visible of fld quizscript to false
    set name of bg button "hide status" to "Show Status"
  end if
end mouseUp


** BKGND #1, BUTTON #13: NEXT GRAM **********************************
on mouseUp
  global count,grams,nextgram,guest,scriptvar
  lock screen
  if count > (the number of lines in grams) then put 1 into count
  put item 3 of line count of grams into nextgram
  put fld question into quest
```

```
   go card id nextgram
   put "Gram" && count into fld num
   put "Gram" && count & ":" & return after fld Quizscript
   put "Gram" && count & ":" & return after scriptvar
   add 1 to count
   put quest into fld question
   unlock screen
end mouseUp

** BKGND #1, BUTTON #14: exit *************************************
on mouseUp
   go "whale main menu"
end mouseUp
```

REFERENCES

1. Antonopoulos, Dionisios G. and Turner, Henry V., *Design and Implementation of the PMS Module for "ARGOS"*, Masters Thesis, Naval Postgraduate School, Monterey, Ca., December 1988.

2. Prina, L.E., "The Silence Deepens", *Sea Power* vol 32, no 7, July 1989

3. Dumas, Ben, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Prentice Hall, Inc., 1988.

4. Shneiderman, Joseph S., *Designing User Interface For Software*, Addison-Wesley Publishing Company, Inc., 1987.

5. Waite, M., and others, *The Waite Group's Tricks of the Hypertalk Masters*, Hayden Books, 1989.

6. Ellis, Richard, *The Book of WHALES*, Alfred A. Knoff, Inc., 1985.

7. Naval Postgraduate School Report No. NPS52-87-046, *Database Computers As Inference Engines*, by David K. Hsiao and Robert B. McGhee, Monterey, Ca, October 1987.

BIBLIOGRAPHY

Anopoulos, Dionisios G. and Turner, Henry V., *Design and Implementation of the PMS module for "ARGOS"*, Masters Thesis, Naval Postgraduate School, Monterey, Ca., December 1988.

Dumas, Ben, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Prentice Hall, Inc., 1988.

Ellis, Richard, *The Book of WHALES*, Alfred A. Knoff, Inc., 1985

Hassab, J.C., *Underwater Signal and Data Processing*, CRC Press, 1989.

Naval Postgraduate School Report No. NPS52-87-046, *Database Computers As Inference Engines*, by David K. Hsiao and Robert B. McGhee, Monterey, Ca, October 1987.

Monk, Andrew, *Fundamentals of Human-Computer Interaction*, Acadedmic Press Limited, 1989.

Prina, L.E., "The Silence Deepens", *Sea Power* vol 32, no 7, July 1989

Shneiderman, Joseph S., *Designing User Interface For Software*, Addison-Wesley Publishing Company, Inc., 1987.

Stephens, R.w., *Underwater Acoustics*, Wiley-Interscience, 1970.

Waite, M., and others, *The Waite Group's Hypertalk Bible*, Hayden Books, 1988.

Waite, M., and others, *The Waite Group's Tricks of the Hypertalk Masters*, Hayden Books, 1989.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center   2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 0142   2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Office of Research Administration   1
   Code 012
   Naval Postgraduate School
   Monterey, California 93943-5002

4. Chairman, Computer Science Dept. (Code 52Mz)   1
   Computer Science Department
   Naval Postgraduate School
   Monterey, California 93943-5002

5. Chief of Naval Research   1
   800 N. Quincy Street
   Arlington, Virginia 22217-5000

6. Center for Naval Analyses   1
   4401 Ford Avenue
   Arlington, Virginia 22302-0268

7. Naval Ocean Systems Center   2
   (Code 72- 1 copy)
   271 Catalina Boulevard
   San Diego, California 92152-5000

8. Curriculum Officer   1
   Computer Technology Program, code 37
   Naval Postgraduate School
   Monterey, California 93943-5000

9. Professor C. Thomas Wu (Code 52Wq)   10
   Computer Science Department
   Naval Postgraduate School
   Monterey, California 93943-5000

10. Maria M. Jamini-Ramirez   2
    Division Head
    MDS Division
    Data Systems Department
    Naval Weapons Station
    Concord, California 94520-5000

11. Robert Calogero   2
    Director SEA CEL-PA
    Logistics Policy and Appraisal Division
    Naval Sea Systems Command
    Washington, D.C. 20362-5101

12. Clifford G. Geiger   1
    Deputy Chief Engineer-Logistics
    Logistics Policy and Appraisal Division
    Naval Sea Systems Command
    Washington, D.C. 20362-5101

13. Lieutenant Deborah R. Kern   2
    1617 Sinking Creek Dr.
    Virginia Beach, Virginia 23464