METHODOLOGY FOR THE CONVERSION OF TACTICAL BOARD GAMES TO COM-ETC(U)

SEP 80 W C GRIGGS, J A THOMPSON
THESIS

A METHODOLOGY FOR THE CONVERSION
OF TACTICAL BOARD GAMES TO
COMPUTER ASSISTED WAR GAMES.

by

William Clifton Griggs
and
John Arthur Thompson Jr.

September 1980

Thesis Advisor: E. Roland

Approved for public release; distribution unlimited
A Methodology for the Conversion of Tactical Board Games to Computer Assisted War Games

William Clifton Griggs and John Arthur Thompson Jr.

Naval Postgraduate School
Monterey, California 93940

Naval Postgraduate School
Monterey, California 93940

Naval Postgraduate School
Monterey, California 93940

Approved for public release; distribution unlimited

As part of the development of the capabilities of the Command, Control and Communications (C3) Laboratory at the Naval Postgraduate School, there was a need for an in-house computer assisted tactical war game. The objective of this thesis was to satisfy that need. An anti-submarine warfare board game, UP SCOPE!, which was designed, developed and produced by Simulations Productions, Inc. of New York, was
(Continuation of abstract)

used as the model. This thesis provides an interactive computer assisted anti-submarine warfare war game called Up Scope which is written in FORTRAN. This thesis also develops a framework for any future computer adaptation of a tactical board game, details a players manual and gives full documentation of the computer programming. Program listings, a sample game and several tactical scenarios are also included.
A Methodology for the Conversion
of Tactical Board Games to
Computer Assisted War Games

by

John Arthur Thompson Jr.
Captain, United States Air Force
B.S., United States Air Force Academy, 1973
M.S., Troy State University, 1978

and

William Clifton Griggs
Lieutenant Commander, Supply Corps
United States Navy
B.S., Miami University, 1971

Submitted in partial fulfillment of the
requirements for the degree of
MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September, 1980

Author

Author

Approved by:

William C. Griggs

John A. Thompson Jr.

Helen F. Roland

Joseph R. Sartora

E. T. Newell Acting

Dean of Information and Policy Sciences
ABSTRACT

As part of the development of the capabilities of the Command, Control and Communications (C3) Laboratory at the Naval Postgraduate School, there was a need for an in-house computer assisted tactical war game. The objective of this thesis was to satisfy that need. An anti-submarine warfare board game, UP SCOPE!, which was designed, developed and produced by Simulations Productions, Inc. of New York, was used as the model. This thesis provides an interactive computer assisted anti-submarine warfare war game called Up Scope which is written in FORTRAN. This thesis also develops a framework for any future computer adaptation of a tactical board game, details a players manual and gives full documentation of the computer programming. Program listings, a sample game and several tactical scenarios are also included.
TABLE OF CONTENTS

I. INTRODUCTION ---------------------------------------- 7

II. DESCRIPTION AND PLAY OF UP SCOPE ----------------- 10
   A. INTRODUCTION ------------------------------------ 10
   B. GENERAL COURSE OF PLAY -------------------------- 11
   C. GAME EQUIPMENT -------------------------------- 12
   D. DEFINITION OF TERMS ----------------------------- 14
   E. GAME INITIATION -------------------------------- 15
   F. COMMAND PHASE ---------------------------------- 16
   G. COMBAT PHASE ------------------------------------ 20
   H. MOVEMENT PHASE ---------------------------------- 26
   I. SEARCH AND DETECTION PHASE --------------------- 27
   J. VICTORY DETERMINATION PHASE --------------------- 32
   K. STATUS DISPLAY PHASE ----------------------------- 32

III. USER'S MANUAL --------------------------------------- 33
   A. INTRODUCTION ------------------------------------ 33
   B. PREGAME REQUIREMENTS ---------------------------- 33
   C. LOADING GAME PARAMETERS AND PLATFORMS ----------- 37
   D. GAME TURNS -------------------------------------- 42
   E. INPUT OF INSTRUCTIONS ----------------------------- 51

IV. PROGRAM DOCUMENTATION ------------------------------- 54
   A. INTRODUCTION ------------------------------------ 54
   B. DESCRIPTION OF MAJOR ARRAYS --------------------- 54
   C. DESCRIPTION OF MAJOR VARIABLES ------------------ 67
   D. DIMENSION, COMMON AND DATA STATEMENTS --------- 71
I. INTRODUCTION

All war games are based on some form of a model, and so it is natural that a research effort concerning war gaming be directed toward the development of a model suitable for that purpose. However, this thesis does not develop a model, basically for two reasons. The first is the fact that the authors' interests are in computer applications. The second is the great wealth of models and modelling technology available today. James F. Dunnigan, the president of Simulations Publications Inc. (SPI), writes, "There are two game-design rules that control all others...Keep it simple... (and)...plagiarize." When he says "plagiarize" he means whenever possible use existing technology for there is no reason to recreate what has been done before. The authors took Mr. Dunnigan at his word, and selected an SPI board game entitled "UP SCOPE!" as the war game on which to base their work.

This research effort has two major objectives. The first is to develop an "in house" war gaming capability for the Command, Control and Communications (C3) Laboratory located at the Naval Postgraduate School. This capability is necessary because it is not sufficient for military students to know what types of games exist, the objectives they are expected to accomplish and how to play them. The students
must also be given the opportunity to gain firsthand knowledge about the design, development and programming of war games. The second objective is to examine the problems associated with the conversion of a tactical board game to a "user-friendly" computer-assisted war game. The thesis develops a general outline to follow when a conversion is planned, and discusses the problems encountered during the conversion of "UP SCOPE!." 

Chapter II, Description and Play of Up Scope, provides the reader with the background, rules and procedures necessary to play the game. Chapter III is a step by step "player's guide" to the computer version of "Up Scope!." It leads the player through the game by including and discussing key portions of a practice game. The entire practice game is included as an Appendix. Chapter IV is titled "Program Documentation," and includes detailed documentation of the FORTRAN statements. The program itself uses standard FORTRAN IV, and is presently operating on an IBM 360 computer. The only change required to transport the program is the inclusion of a random number generator subroutine called RANDOM. The IBM 360 located at the Naval Postgraduate School has such a subroutine in its library. It is hoped that Chapter IV will make it easy for the program to become the basis for future research and expansion by other students having an interest in "grass roots" war gaming. Chapter V presents the outline developed for conversion of a board
game to a computer-assisted game, and it discusses the problems encountered and changes made to "UP SCOPE" during the conversion.

It is believed that the outline and discussion of problems encountered included in the thesis and actual play of the computerized version of "UP SCOPE!" will greatly increase the military students' awareness of the gaming, and give him a better understanding of what designing, developing and programming a war game entails.
II. DESCRIPTION AND PLAY OF UP SCOPE

A. INTRODUCTION

Up Scope is essentially a computer aided, interactive, tactical simulation of submarine and anti-submarine warfare in the 1980's. It has been adapted from the tactical simulation board game "UP SCOPE!" which was designed, produced and marketed by Simulations Publications Inc., New York, N.Y. "UP SCOPE!" is used with SPI's permission.

This war game furnishes fairly complete, albeit unclassified, information on every significant submarine and anti-submarine platform and weapon system employed by both the NATO powers and the Soviet Union today. Although this game is essentially designed for two players (Red and Blue), any reasonable number can play. As in most war games, the additional players would be relegated to subordinate commander roles. It should be noted that when the game is played with two players, the game is rigid-ruled and self-umpiring. Since this thesis is unclassified, the actual performance parameters for weapon systems, sensors, aircraft, ship and submarine characteristics were derived from unclassified sources and adapted in conjunction with parameters currently implemented in the board game "UP SCOPE!"
B. GENERAL COURSE OF PLAY

Up Scope is basically designed for two players, one of whom normally commands the submarine force while the opposition commands the anti-submarine force. After deciding upon the particular scenario to be played, each player selects his force composition. Appendices B, C and D contain the relevant Ship, Weapon and Aircraft Data Summaries. The following table illustrates the number of surface ship and submarine classes available for selection from each represented country.

<table>
<thead>
<tr>
<th>Country</th>
<th>Number of Surface Ship Classes</th>
<th>Number of Submarine Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Canada</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>France</td>
<td>16</td>
<td>5</td>
</tr>
<tr>
<td>Great Britain</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>Italy</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>Soviet Union</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>United States</td>
<td>28</td>
<td>11</td>
</tr>
<tr>
<td>West Germany</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

Up Scope is played in a series of successive game turns, each of which consists of a command phase, a combat phase, a movement phase, a search phase, a victory determination phase and a display phase. During the command phase each player privately determines combat, movement (speed and
destination) and search missions for all units under his control.

As this is essentially an anti-submarine warfare game, the only sensors employed are active and passive sonar suits. Radar, ESM and visual detection methods are not pursued.

At the end of the command phase all commands have been entered on the keyboard. The computer then executes combat, movement and search missions for every platform involved. Although the program executes combat, movement and search missions sequentially, they are considered and treated as though they occur simultaneously.

C. GAME EQUIPMENT

The game board is a numbered hexagonal grid pattern superimposed over open ocean. Each "hex" represents a one nautical mile area of ocean. Thus, traversing four hexes is the equivalent of moving four nautical miles. Each hex address is composed of a four-digit number (i.e. XXYY); the XX represents the column coordinate and the YY represents the row coordinate. Figure 1 illustrates the four corners and middle section of the game board.

During the play of Up Scope, reference is made to "adjacent" hexes. Any hex that has a common border with another hex is an adjacent hex. Therefore, each hex except those located on the boundaries has six adjacent hexes. Referring to Figure 1, the hexes containing a $ are adjacent to hex 1622.
Figure 1. Game Board
D. DEFINITION OF TERMS

1. **Platform Class**

   The Ship Data Summary, Appendix B, categorizes all surface ships and submarines by class. The characteristics of any particular platform are located in Appendix B according to the platform's class. For instance, the Charlie and Papa class Soviet submarines share the same characteristics, and are located in the Ship Data Summary under the Soviet submarine class entitled "Charlie".

2. **Speed Command**

   During the command phase, each platform is assigned a speed command which indicates a rate of speed at which the platform is considered to be moving during the current game turn. The speed command determines the total number of hexes the platform must move during the movement phase of the game turn.

3. **Search Command**

   During the command phase, each eligible platform may be assigned a search command indicating whether the active sonar, passive sonar or both active and passive sonar suits are to be employed during the search phase of the game turn.

4. **Search and Detection**

   Detections by a platform's active sonar, passive sonar or by both will be recorded for an aircraft or surface ship only against submarine targets, and for a submarine against both surface ships and submarines. Detections
presented will include friendly as well as adversary targets. It is up to the player to discriminate which target is which. In general a platform may only attack a platform it detected during the prior game turn.

5. Detection Value
When executing search commands a searching platform is assigned a detection value which indicates the basic ability of its detection equipment to detect and localize previously undetected targets.

6. Evasion Value
When a platform executes a search command all potential targets are assigned an evasion value. This value indicates the platform's basic ability to avoid detection based on both the platform's inherent characteristics and/or range, ocean and water conditions which aid the platform's ability to evade detection.

E. GAME INITIATION
Appendix A lists several scenarios, each of which constitutes a distinct game. Players may select any of those scenarios or develop their own scenarios as they desire. Initializing the game requires each player to select a password which enables the player to enter commands and retrieve relevant information regarding combat operations, damage reports and search results of platforms under his command. The player then selects his force composition (i.e. three surface ships and two submarines), gives each
platform a name, identifies the platform class (Spruance, Nimitz, Krivak, Charlie, etc.) and designates the initial position on the game board. After entering this initial information the game is ready to begin. The game consists of game turns, each of which represents six minutes of elapsed time. Game turns are executed independently, the only exception is the fact that a scheduled combat mission can succeed only if that particular platform detected the target on the previous game turn. Game turns are then executed until the game ends, either by preset victory conditions being met, the preselected number of game turns having been executed or mutual decision by both Red and Blue players. Figure 2 displays a block diagram of the phases of Up Scope.

The availability of aircraft is restricted to ship-based aircraft carried aboard the combatants involved in the scenario. Both fixed wing aircraft and helicopters are available. The Ship Data Summary, Appendix B, lists type and number of aircraft carried aboard the various classes of surface ships. The Aircraft Data Summary, Appendix D, lists the detection values, maximum speed, weapon system mix and resupply ratings for each type aircraft. Some aircraft may carry either of two different weapon systems. The player must decide before launching the aircraft which weapon system he desires to employ. An aircraft weapon load cannot be changed after launch.

F. COMMAND PHASE

After reviewing the previous game turn’s displays of combat action results, damage suffered by friendly forces,
Scenario Selection and/or Design

Force Composition Selection

Initializing and Force Loading

Command Input Phase

Combat Phase

Movement Phase

Search Phase

Victory Determination Phase

Status Display

Figure 2. Up Scope Block Diagram
current platform locations and detections resulting from the previous game turn searches, the player privately determines his current game turn strategy.

1. **Combat Operations**

   Combat operations are subdivided into two categories: Aircraft combat operations and surface ship/submarine combat operations.

   Aircraft can only release one weapon system per game turn and only at a submarine target. The player selects either weapon system number one or weapon system number two. The weapon systems carried are defined in the Aircraft Data Summary, Appendix D. The weapon system listed first is weapon system number one and the system listed second is weapon system number two. Therefore, to engage a target, the player enters weapon system number one or two and the target's hex address location. Aircraft cannot be attacked during any scenario under the present rules of Up Scope.

   Surface ships and submarines may engage a single target with one or more of their weapon systems simultaneously; therefore, the player may enter more than one combat mission for each platform in the manner described above. Due to the nature of the game, there is no interference between weapons in a multiple weapon system attack. The weapon systems carried by each platform are listed in the Ship Data Summary, Appendix B. Weapon system numbering is in the order they are listed in Appendix B.
2. **Movement**

The player must keep in mind that the number of hexes a platform traverses during each game must be consistent with the current maximum speed allowed and that the speed must equal the distance to the destination hex. The player enters projected speed and destination hex address each game turn.

At the start of a scenario, all aircraft are considered aboard their respective surface ships and may not move independently until they are launched. During each command phase, a player may launch any single aircraft from each capable surface ship under his control. After launch the aircraft move according to the established movement rules. Aircraft may not be launched once it has been determined the parent ship is sunk. Also, if a parent ship departs the board boundary, the aircraft must immediately depart the board boundary.

3. **Search**

The player may not assign a search mission to any aircraft that is either not flying (still based on the parent ship) or is transiting to a new destination hex. To execute a search mission the aircraft must have a speed of zero for that game turn, which signifies that a fixed wing aircraft is loitering in the area or that a helicopter is either hovering or loitering in the area. Additionally, aircraft may not be assigned both active and passive search missions simultaneously.
For surface ships and submarines, only platforms with a one in the column labeled "SS" in the Ship Data Summary, Appendix B, may execute both active and passive search missions simultaneously. The player, when asked in the game turn input phase, enters a zero for no search mission, a one for an active search mission, a two for a passive search mission and a three for both active and passive search missions simultaneously. If a platform is not assigned a detection value under active and/or passive detection in the Ship Data Summary, it may not be assigned a search mission employing that detection capability. For example, the Nimitz class aircraft carrier possesses neither type of detection value and thus cannot execute a search command. Aircraft may not be assigned a search mission if the aircraft is assigned a move command other than "none."

The assignment of search commands is at the player's discretion. A platform is never required to execute a search mission.

G. COMBAT PHASE

After completion of the command phase and players have input their combat, movement and search commands, the program shifts into execution. The first phase is the combat phase which executes all combat orders.

1. Introduction

Contemporary weapon systems include a large variety of ship-to-ship missiles, sophisticated wire-guided and/or
homing torpedoes, missile launched depth bombs/torpedoes and depth charges. The Ship Data Summary, Appendix B, along with the Weapon Data Summary, Appendix C, indicate all the following information.

a. Weapon type

The various weapon types with which each platform class is equipped. Surface ships and submarines carry a maximum of three different weapon systems.

b. Accuracy Rating

Each weapon type has an accuracy rating ranging from six (best) to three (worst).

c. Maximum Range

Each weapon system has a maximum effective range expressed in terms of number of hexes it will traverse.

d. Weapon Resupply

The number of weapon delivery units available for each weapon system, ranging from one to six. A weapon delivery unit is composed of the number of individual weapons normally expended in a single attack utilizing that weapon system. For example, if a Charlie class submarine would normally expend all eight SSN-7 missiles in an attack, then the number of weapon delivery units would be one, and it would be composed of eight SSN-7 missiles.

e. Defense Rating

The Defense rating of each surface ship/submarine class, range from three (best) to zero (worst). It is a
measure of how capable the platform is in defending against
attacks and/or evading the attacks.

f. Aircraft

The number and type of aircraft each surface ship
class carries and is capable of employing in an anti-submarine
role.

2. Execution of Combat Missions

Combat missions are executed during the game turn
according to the following restrictions.

a. Resolution

Each combat mission is resolved separately
against the target platform using the procedure given below.

b. Attacking Platform

Each weapon system type may be employed only once
during a game turn. Different weapon systems aboard an
attacking surface ship/submarine may only be employed against
the same target. (i.e., a platform may not attack two or
more targets during the same game turn).

c. Target

An attacking platform must have detected the
target on the previous game turn. A weapon may be employed
against a undetected target; however, a hit cannot be
obtained.

d. Damage

All combat is considered to be simultaneous.
All damage inflicted occurs at the end of the game turn and
does not affect current game turn commands.
e. Resupply

Once all reloads of a weapon system have been expended, the weapon system cannot be employed. There is no external resupply of any weapon system during a game.

3. Resolution of Combat Actions

Damage inflicted due to combat resolution takes effect at the end of the game turn; thus, a platform that has been sunk or hit is able to execute and resolve all assigned combat missions. For each combat mission the program checks to insure the target is within weapon range, cross references the accuracy rating of the employed weapon with the defense rating of the target platform, draws a random number to simulate rolling a die and locates the resulting random number (an integer from one to six) under the proper resolution column on the Combat Results Table, Figure 3. If the indicated result is not an "H" (hit), then the attack failed. If the result is an "H", then the attack was successful and the durability value of the target is reduced by one.

4. Weapon System Resupply

Each weapon system carried by a platform has a weapon system resupply rating which indicates the comparative supply and reload facilities available for that weapon system on that platform. Each time a given platform executes a combat mission, the corresponding resupply rating for that weapon system is decremented by one. Once a resupply rating is
### Defense Rating of Target Platform

<table>
<thead>
<tr>
<th>Weapon</th>
<th>6</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Rating</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>any</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>H</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Random</td>
<td>2</td>
<td>H</td>
<td>H</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Number</td>
<td>3</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Drawn</td>
<td>4</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>

Figure 3. Combat Results Table
reduced to zero that platform may not be assigned a combat mission employing that weapon system.

5. **Platform Damage**

The durability value assigned to each surface ship/submarine indicates the number of hits required to sink it. A submarine which has a durability value of two would require two separate hits to sink it. The initial durability value for each type of platform is indicated below.

<table>
<thead>
<tr>
<th>Type of Surface Ship/Submarine</th>
<th>Initial Durability Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft carrier</td>
<td>3</td>
</tr>
<tr>
<td>Cruiser, Submarine</td>
<td>2</td>
</tr>
<tr>
<td>Destroyer, Frigate</td>
<td>1</td>
</tr>
</tbody>
</table>

a. **Damage Assessment**

For each durability point of damage suffered, the platform's maximum speed, weapon system resupply ratings, active detection value and passive detection values are reduced by one.

b. **Repair**

Damage may never be repaired during the scenario.

c. **Sunk**

When a platform's durability point value is reduced to zero, the platform is considered sunk.
6. Disturbed Water

Whenever a combat mission results in a weapon detonation, disturbed water is assumed to occupy the hex in which the target is residing and the six adjacent hexes as well. Detonations will not occur when a weapon system misses the target and the weapon is a torpedo; however, depth charges and depth bombs will detonate regardless of a hit or miss. The effects of disturbed water are discussed later in this chapter.

H. MOVEMENT PHASE

After completion of the combat phase during which all the combat orders are executed, the program shifts to the movement phase and all movement commands are executed.

The speed command given a platform during the command phase determines the total number of hexes it must traverse during the current game turn. It also must match the hex distance between the current location and future location. Any platform may be assigned a speed of zero for any game turn. Aircraft must have a speed of zero in order to conduct a search mission. Also surface ships and submarines, in general, have better detection capabilities at lower speeds than at higher speeds.

The maximum speed is limited to the maximum for that class of platform as indicated in the Ship Data Summary, Appendix B. Also, the maximum speed will be permanently
reduced as a result of damage sustained in combat operations.

I. SEARCH AND DETECTION PHASE

After completion of the movement phase, the program shifts to the search and detection phase, during which the search commands previously entered are executed for all platforms.

1. Introduction

All platforms (surface ships, submarines or aircraft) in this game employ only detection capabilities which simulate the use of both active and passive advanced sonar equipment. The detection and evasion capabilities for all platforms are delineated in the Ship and Aircraft Data Summaries in appendices B and D respectively. This includes the following:

a. Active Detection Value

The platform’s active detection value (ADV) simulates the capability of its active sonar system(s). The ADV ranges from five (best) to one (worst).

b. Passive Detection Value

The platform’s passive detection value (PDV) simulates the capability of its passive sonar system(s). The PDV ranges from five (best) to three (worst).

c. Evasion Value

Surface ship’s and submarine’s evasion values (EV) which simulate the platforms radiated noise level. The
higher the EV value, the quieter the platform is considered to be and therefore the more difficult it is to locate using a passive search during the game turn. The EV ranges from five (best) to one (worst).

d. Simultaneous Active and Passive Search Missions

If a platform has a one in the "SS" column for its class in the Ship Data Summary, Appendix B, then the platform is capable of executing both active and passive search missions simultaneously during the game turn.

e. Speed Degradation

Normally, sonar system performance is degraded the faster the surface ship or submarine is traveling, due to increased water flow noise and the own ship's radiated noise level. Some more advanced sonar systems have solved this degradation to some degree. In Up Scope if a platform has a one in the "SD" column for its class in the Ship Data Summary, Appendix B, this indicates that when a platform executes a passive search mission there is neither degradation in capability nor reduction due to platform speed in the passive detection value.

2. Search Mission Resolution

Search missions assigned during the command phase are resolved during this search and detection phase. The program computes the appropriate active or passive detection value of the searching platform, taking into account all the appropriate modifiers, then calculates evasion values for all other
platforms except aircraft. The program then compares the evasion value of each platform to the final adjusted detection value of the searching platform. If the detection value is greater than or equal to the evasion value, the target platform is considered detected by the searcher. If the evasion value is greater than the detection value then no detection has been made.

a. Active/Passive Detection Value Computation

When a platform executes a search mission, its appropriate detection value (ADV or PDV) is cumulatively modified by the following criteria:

(1) The platform's ADV/PDV is reduced by one if its speed is one or two.

(2) The platform's ADV/PDV is reduced by two if its speed is three or four.

(3) The platform's ADV/PDV is reduced by two if the water condition has been set as "erratic," discussed later in this chapter.

(4) The platform's ADV/PDV is reduced by three if the platform occupies a disturbed water zone, discussed later in this chapter.

(5) The platform's ADV/PDV is reduced by one for each durability point of damage sustained by the platform.

(6) As already mentioned, the platform's ADV/PDV is not reduced if the platform's speed is zero, or if column "SS" for the class in the Ship Data Summary, Appendix B, has a one on it. Otherwise, the ADV/PDV is reduced as stated above.
(7) A platform's ADV/PDV cannot be reduced below zero.

b. Active Evasion Value (AEV) Computation

For every active search mission being executed, the program computes an AEV for every potential target in the following fashion (This computation does not make use of the evasion value listed in the Ship Data Summary, Appendix B.):

(1) The basic AEV is determined by computing the hex distance between the two platforms, dividing the result by three and rounding up. The following Table illustrates the computation.

<table>
<thead>
<tr>
<th>AEV</th>
<th>Hex Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 - 3</td>
</tr>
<tr>
<td>2</td>
<td>4 - 6</td>
</tr>
<tr>
<td>3</td>
<td>7 - 9</td>
</tr>
</tbody>
</table>

(2) The basic AEV is increased by three if the platform occupies a disturbed water zone.

c. Passive Evasion Value (PEV) Computation

For every passive search mission being conducted, the program computes a PEV for every potential target in the following fashion:

(1) The platform's basic PEV is determined by computing the hex distance between the two platforms, dividing the result by ten and then rounding up. The following illustrates the computation:
3. Water Conditions

During any scenario, the water conditions are prescribed by the scenario rules as either "Normal" or "Erratic" and this information is set during the initializing and force loading phase. It should be recognized that if the water conditions are erratic during a scenario, the effectiveness of both active and passive sonar suits is degraded.

Disturbed water zones are created by the detonation of weapons. The zone consists of the hex in which the weapon

<table>
<thead>
<tr>
<th>PEV</th>
<th>hex distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 - 10</td>
</tr>
<tr>
<td>2</td>
<td>11 - 20</td>
</tr>
<tr>
<td>3</td>
<td>21 - 30</td>
</tr>
</tbody>
</table>

(2) The platform's PEV is increased by the EV indicated for its class in the Ship Data Summary, Appendix B.
(3) The platform's PEV is increased by three if it occupies a disturbed water zone.
(4) The platform's PEV is decreased by two if the platform is assigned an active search mission for the current game turn.
(5) The platform's PEV is decreased by three if the platform executed a combat mission during the game turn.
(6) The platform's PEV is automatically set at six if the platform remained stationary, was not assigned an active search mission and did not execute a combat mission during the current game turn.
detonated and the six adjacent hexes. The disturbed water zones exist only during the current game turn. If occupied by a searching platform the zone decreases detection values and if occupied by potential targets, the evasion value is increased as just described.

J. VICTORY DETERMINATION PHASE

After completing the search phase, the program enters the victory determination phase. If all of either player's ship forces have been sunk, the program exists after displaying the final results. If either the predetermined number of ships designated during the initializing and force loading phase have been sunk, or the designated number of game turns have been completed, the program attempts to end the game. It queries the players to see if they desire to continue or, in fact, desire to exit the game.

K. STATUS DISPLAY PHASE

During the status display phase the player reenters the game by requesting the various status displays available regarding damage inflicted, damage suffered, search results and individual platform status. Then the player enters the command input phase.
III. USER'S MANUAL

A. INTRODUCTION

The computerized version of Up Scope is an easy game to play even without a complete understanding of the rules discussed in Chapter II. All the difficult chores, such as computation of detections and damage, and all bookkeeping tasks are performed by the computer. Through the interactive nature of the game, a player is led through the various stages, giving all the information the player needs to know to make decisions. However, a general understanding of the rules and game procedures is required to use the board and Appendices. An understanding of the game also allows the players to play more effectively.

This chapter includes a section on pregame requirements which deals with the development of scenarios. It also includes several sections containing the instructions needed to use the computer program. The instructions are based on excerpts from the practice game in Appendix E. Instructions for obtaining a copy of the program deck are contained in Appendix G.

B. PREGAME REQUIREMENTS

The pregame requirements for Up Scope can be as simple or as complicated as the players desire. However, as a
minimum, the following steps in developing a scenario should be taken to ensure a successful game.

1. **Objective**

   Every game played should have an objective. This could mean a detailed write-up, with a general scenario to set the world's political and economic situation and a specific scenario setting the tactical situation. It could also be a simple agreement between the players to "search and destroy." Whatever the purpose, both players should agree to the basic objective of the game before they start.

   The objective should also include victory determination criteria. This is extremely important because it forces the players to concentrate on setting and achieving the objective. Again the criteria could be simple, such as "fight until one force is totally destroyed", or complex. A complex criteria might be a force level percentage after a specified number of game turns. Appendix A presents several scenarios modified from examples included in the SPI version of "UP SCOPE!".

2. **Game Environment**

   Another important aspect of the scenario is the environment in which the game is to be played. In the computerized version of Up Scope, this includes the size of the ocean area and the water conditions.
a. Game Board

In the board game, SPI supplies the players with two 33 by 45 hex boards. The boards can be placed side by side or lengthwise to increase the size of the playing area. The available size in the computer game is physically unrestricted as long as the number of hexes in the X or Y direction does not exceed 99. When determining the size, the objective of the game and the speed of the ships must be considered. For example, it would take the fastest ship 11 turns just to cross the 33 hex width of the SPI board. Unless the players are willing to spend a great deal of time maneuvering or the scenario dictates otherwise, the size of the area should be held to a minimum.

Up Scope does not restrict the players from moving their forces outside the agreed upon playing area. A "gentlemen's agreement" between the two players must exist to forestall "out of bounds movement."

b. Water Condition

The second environmental factor in Up Scope is the water condition which was discussed in chapter II.I.3. The water condition can be "normal" or "erratic."

3. Game Length

The game can be terminated by any of three methods. Two of them, game turns and number of ships destroyed, are scenario dependent. The third is total destruction of a player's ship forces.
a. Game Turns

The number of game turns to be played is mutually agreed on by the players. It can be used in two ways. The first is actually determined by the scenario, which might involve the Blue forces defending a certain area for a given number of turns. The second use is to put an upper limit on the length of the game. In either case it is preferable to use a relatively small number because the players are given the option of continuing the game beyond the preselected number of game turns.

b. Disengagement

Both players are given the opportunity to prematurely end the game by disengaging when a certain number of their ships have been destroyed. Again, a relatively small number should be chosen because the player will be given the option of continuing when his disengagement criterion has been reached.

c. Total Destruction

Total destruction of a player's forces is the only means of terminating the game without the player's consent. When this occurs, the program will print the final results, and give the players an opportunity to start a new game.

4. Forces

Perhaps the major pregame task, other than determination of the objective, is to decide the forces to
be employed by the Blue and Red players. The number and structure of the forces should be consistent with the objective. Normally the forces should be balanced between the players so that the outcome is determined by tactics and not by superior numbers.

Ship Data Summary, Appendix B, contains all the ship classes in the Up Scope arsenal and all the required information about each class. The Appendix must be consulted when determining the number of ships because it contains the number of aircraft carried by each class. This is important because the current version of the game can only support 100 platforms, including aircraft, and the game loads aircraft automatically for each ship class. Prior to accessing the game, the players should ensure that their combined number of aircraft, surface ships and submarines does not exceed 100.

C. LOADING GAME PARAMETERS AND PLATFORMS

The loading portion of Up Scope consists of several sections; each designed to ease the entry of game inputs.

1. Menu Selection

The first display which the players see when Up Scope is accessed is:

EXECUTION BEGINS...

THE FOLLOWING OPTIONS ARE AVAILABLE TO YOU:

1. ZERO ANY PREVIOUSLY ESTABLISHED DATA BASE.
2. EXIT THE GAME.
3. INITIALIZE GAME PARAMETERS AND COMMENCE PLAY.
ENTER AN OPTION.

>3

Option 1 reinitializes any user inputs from previous games, which allows the players to start a new game without exiting and restarting the program. Options 2 and 3 are self-explanatory. If Option 1 is selected, the players will automatically be returned to menu selection at its completion.

2. Entry of Game Parameters
   a. Basic Parameters

   Once Option 3 has been selected the program will begin to ask for the basic game parameters. First, the following warning concerning the total number of platforms is printed with a request for the number of game turns.

   YOU ARE NOW GOING TO ENTER THE BASIC GAME PARAMETERS.
   ONE WORD OF CAUTION. THE PRESENT GAME CAN ONLY HANDLE A TOTAL OF 100 WEAPON SYSTEMS. INCLUDED IN THESE SYSTEMS ARE SURFACE SHIPS, SUBMARINES AND AIRCRAFT. THE NUMBER OF AIRCRAFT ARE DETERMINED BY THE SHIP CLASS AS REFLECTED IN THE DATA BASE, AND WILL BE LOADED AUTOMATICALLY WHEN YOU SELECT THE SHIP CLASS. PRIOR TO BEGINNING THE LOADING PROCEDURE YOU SHOULD ENSURE THAT THE TOTAL NUMBER OF EACH PLAYERS WEAPON SYSTEMS DOES NOT EXCEED 50.

   ENTER THE NUMBER OF GAME TURNS YOU WANT TO PLAY.
   >2

   After game turns have been entered, the program will print the following request for the water condition.

   ENTER THE WATER CONDITION (0-NORMAL, 1-ERRATIC).
   >0
b. Blue/Red Game Parameters

The following lines begin the entry of the Blue player's game parameters by requesting a Blue password. This player has selected "BLUPAS".

THE FOLLOWING INPUTS APPLY ONLY TO THE BLUE PLAYER
THE FIRST IS THE BLUE PLAYERS PASSWORD WHICH IS
A MAXIMUM OF 6 CHARACTERS. THE PASSWORD IS IMPORTANT BECAUSE
IT IS REQUIRED PRIOR TO THE INPUT OF COMBAT, MOVE AND SEARCH
COMMANDS. IF YOU FORGET THE PASSWORD, YOU WILL HAVE TO RETURN
TO MENU SELECTION, AND RESTART THE GAME.

SELECT AND ENTER YOUR PASSWORD
> blupas

YOUR PASSWORD WILL BE -BLUPAS- . IF YOU ARE
READY TO CONTINUE TYPE Y(ES). TYPE N(O) IF YOU WISH TO
CHANGE YOUR PASSWORD.
> y

Next the program will ask for the number of Blue surface ships and submarines, and the number of ships that must be destroyed for disengagement to occur. The format as seen below echoes the inputs and gives the player the opportunity to change his choices.

ENTER THE NUMBER OF BLUE SURFACE SHIPS.
> 2

ENTER THE NUMBER OF BLUE SUBMARINES.
> 1

ENTER THE NUMBER OF BLUE SHIPS THAT MUST BE DESTROYED FOR PREMATURE GAME ENDING.
> 2

ECHO
BLUE SURFACE SHIPS - 2  
BLUE SUBMARINES - 1  
NUMBER OF BLUE SHIPS DESTROYED TO END GAME - 2  

TYPE Y(ES) IF THESE VALUES ARE CORRECT, OR N(0) IF YOU WANT TO REENTER THEM.  
> y  

If the Blue player does not elect to change his inputs, the program will advance to the Red player's game parameters which is essentially a repeat of the above. In this practice game, the Red player has entered two submarines of which the destruction of one will cause disengagement (not shown).  

C. Loading Platforms  

Up Scope leads each player through the loading procedure one ship at a time. It will first ask for the player's password. If the password is entered correctly, the player will input a name (six characters or less) for the ship, the ship's class and its initial location. The program automatically loads any aircraft carried by a class of ship, and asks the player for aircraft names. Inputs for the practice game are shown below.  

BLUE SURFACE SHIP NUMBER 1  
ENTER A 6 CHARACTER NAME FOR THE SHIP.  
> tincan  
ENTER THE SHIP CLASS OF TINCAN  
> spruance  
ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF TINCAN  
> 500
In the above example the hex address has been entered as "500" instead of "0500". The program treats the hex address as a four digit integer and pads leading blanks with zero's. In the following example the entry is "0520".

THE SPRUANCE CLASS OF SHIP HAS 1 SEA SPRITE TYPE AIRCRAFT.

ENTER A 6 CHARACTER NAME FOR AIRCRAFT 1
sprul

BLUE SURFACE SHIP NUMBER 2

ENTER A 6 CHARACTER NAME FOR THE SHIP.
escort

ENTER THE SHIP CLASS OF ESCORT
knox

ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF ESCORT
0520

THE KNOX CLASS OF SHIP HAS 1 SEA SPRITE TYPE AIRCRAFT.

ENTER A 6 CHARACTER NAME FOR AIRCRAFT 1
knoxi

BLUE SUBMARINE NUMBER 1

ENTER A 6 DIGIT CHARACTER NAME FOR THE SUBMARINE.
lapon

ENTER THE SUBMARINE CLASS OF LAPON
sturgeon

ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF LAPON
510

YOU ARE NOW READY TO LOAD THE RED WEAPON SYSTEMS.

ENTER THE RED PASSWORD.
redpas

RED SUBMARINE NUMBER 1
ENTER A 6 DIGIT CHARACTER NAME FOR THE SUBMARINE.
> ivan

ENTER THE SUBMARINE CLASS OF IVAN
> charlie

ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF IVAN
> 1510

RED SUBMARINE NUMBER 2

ENTER A 6 DIGIT CHARACTER NAME FOR THE SUBMARINE.
> boris

ENTER THE SUBMARINE CLASS OF BORIS
> november

ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF BORIS
> 1520

D. GAME TURNS

Once the players have entered the game parameters, they are ready to enter the gaming phase.

1. Displays

The first step of each game turn is to review the results of the previous turn and enter combat, move and search instructions. These tasks are controlled by a group of routines labeled STATUS. After the last Red submarine has been loaded, the program will ask for the Blue player's password and print the following menu of STATUS display options:

BLUE CALL TO STATUS ROUTINES. ENTER THE BLUE PASSWORD.
> bluepas
THE FOLLOWING DISPLAY OPTIONS ARE AVAILABLE TO YOU:

1. DETECTIONS OF ENEMY PLATFORMS.
2. RESULTS OF WEAPONS FIRED.
3. DAMAGE TO FRIENDLY PLATFORMS.
4. STATUS OF FRIENDLY PLATFORMS AND INPUT OF INSTRUCTIONS.
5. BEGIN NEXT GAME TURN.

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

>4

If the current game turn is the first turn, the first three options will contain no information, and should be ignored.

a. Detection of Enemy Ships

This option will print the detections of each platform of the calling side in the following format:

<table>
<thead>
<tr>
<th>DETECTING PLATFORM</th>
<th>SONOR USED</th>
<th>TYPE</th>
<th>DETECTED TYPE</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TINCAN</td>
<td>ACTIVE</td>
<td>SUBMARINE</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>ESCORT</td>
<td>ACTIVE</td>
<td>SUBMARINE</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>ESCORT</td>
<td>ACTIVE</td>
<td>SUBMARINE</td>
<td>1522</td>
<td></td>
</tr>
<tr>
<td>TINCAN</td>
<td>PASSIVE</td>
<td>SUBMARINE</td>
<td>1522</td>
<td></td>
</tr>
<tr>
<td>LAPON</td>
<td>PASSIVE</td>
<td>SURFACE</td>
<td>502</td>
<td></td>
</tr>
<tr>
<td>LAPON</td>
<td>PASSIVE</td>
<td>SURFACE</td>
<td>720</td>
<td></td>
</tr>
<tr>
<td>LAPON</td>
<td>PASSIVE</td>
<td>SUBMARINE</td>
<td>1511</td>
<td></td>
</tr>
<tr>
<td>LAPON</td>
<td>PASSIVE</td>
<td>SUBMARINE</td>
<td>1522</td>
<td></td>
</tr>
</tbody>
</table>

END OF DETECTION ROUTINE. TYPE Y(ES) TO CONTINUE.

> y

b. Results of Weapons Fired

Option 2 examines each friendly platform, and gives the probability of a hit for each weapon fired during
the previous turn other than torpedoes. The result for torpedoes is either "DETONATION" or "NO DETONATION". The following is one example from the practice game:

SHIP - ESCORT COMBAT LOCATION - 1522
WEAPON 2 TYPE - MK 46 PROB(HIT)/DETONATION - DETONATION

SHIP - LAPON COMBAT LOCATION - 1522
WEAPON 1 TYPE - SUBROC PROB(HIT)/DETONATION - 0.6667
WEAPON 2 TYPE - MK 48 PROB(HIT)/DETONATION - DETONATION

END OF WEAPON RESULT DISPLAY. TYPE Y(ES) TO CONTINUE.

Some discussion is offered for some of the displays listed above in this section. Above, the Lapon, a sturgeon class nuclear attack submarine, released a MK 48 torpedo and a Subroc against a target located at 1522. The 'WEAPON 1 TYPE-SUBROC' statement illustrates the fact that some weapons detonate regardless of proximity of target, while others (notably torpedoes), only detonate when in close proximity, or when actually hitting a target. In either case, it is classified as a hit. Therefore, when engaging with a "non-torpedo" type of weapon, the player is not sure if the target was hit or destroyed. All that is displayed is the probability of a hit. The player should be aware that torpedo type weapons, on the other hand, either hit (detonate) or miss (no detonation).

c. Damage to Friendly Platforms

This option will give the player the number of hits each ship has suffered if it is not sunk. Otherwise "SUNK" is printed as below:

44
SHIP       STATUS/HITS
TINCAN       0
ESCORT       SUNK
LAPON       0

END OF DAMAGE DISPLAY ROUTINE. TYPE Y(ES) TO CONTINUE.
> y

d. Status of Friendly Platforms and Input of Instructions

This option displays individual platform status and accepts and processes all relevant commands. The program is descriptive and self-prompting. There are differences between aircraft and other platforms; therefore, there are two slightly different routines. Aircraft must be launched from the parent ship, must have a speed of zero to enable search missions and they must only carry one weapon system during a flight.

As this option is entered, the following is displayed:

YOU HAVE SELECTED INDIVIDUAL PLATFORM STATUS DISPLAY AND ASSOCIATED COMBAT, MOVEMENT AND SEARCH COMMAND INPUTS. ENTER THE NAME OF THE DESIRED PLATFORM.

(1) Initial display for non-aircraft platforms

The player will see the following and make the entries in the same fashion as in this example:

> tincan
TINCAN IS LOCATED AT POSITION 500 SPEED 0
IT WAS NOT ASSIGNED A SEARCH MISSION
TINCAN WEAPON SYSTEM(S)

<table>
<thead>
<tr>
<th>TYPE</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MK 46</td>
</tr>
<tr>
<td>2</td>
<td>ASROC</td>
</tr>
</tbody>
</table>

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

> n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

> 2

> 502

ENTER SEARCH MISSIONS AS FOLLOWS:
0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION
3 ACTIVE AND PASSIVE

> 3

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
NONE SCHEDULED
DESTINATION 502 SPEED 2
ACTIVE AND PASSIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION

> y

ENTER THE NAME OF THE DESIRED PLATFORM

The lines starting with the ' > ' are the player's entries.
The player should recognize that when an active search
mission and/or a combat mission is assigned, the platform's
risk of being detected is significantly greater.
(2) Full display for non-aircraft platforms. If a platform had combat and search missions scheduled during the previous game turn, the results of these missions are reported just before new input instructions are asked for. The following illustrates the full range of information displays; i.e., Combat results; damage inflicted; movement and search missions undertaken and subsequent orders for the present game turn:

> lapon

LAPON IS LOCATED AT POSITION 712 SPEED 2

THE LAPON ATTACK ON TARGET LOCATED AT 1522

RELEASE OF SUBROC RESULTED IN A DETONATION WITH PROBABILITY OF A HIT OF 4 OUT OF 6

RELEASE OF MK 48 RESULTED IN A DETONATION (HIT)

IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)

<table>
<thead>
<tr>
<th>ACTIVE</th>
<th>TYPE</th>
<th>LOCATION</th>
<th>METHO</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBMARINE</td>
<td>1</td>
<td>1311</td>
<td>ACTIVE</td>
<td>6</td>
</tr>
<tr>
<td>LAPON WEAPON SYSTEM(S)</td>
<td>TYPE</td>
<td>RANGE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>SUBROC</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>MK 48</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER TARGET LOCATION WEAPON SYSTEM NUMBER IN THIS FASHION

> 1311
> 1

TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER OTHERWISE ENTER N(0)

> 2

TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER OTHERWISE ENTER N(0)

> n
TO SCHEDULE MOVEMENT ENTER SPEED DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

>2
>714

ENTER SEARCH MISSIONS AS FOLLOWS:
0  NONE SCHEDULED
1  ACTIVE MISSION
2  PASSIVE MISSION
3  ACTIVE AND PASSIVE

>1

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
  SUBROC WEAPON TGT LOCATION 1311
  MK 48 WEAPON TGT LOCATION 1311
DESTINATION 714 SPEED 2
ACTIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION

> n

(3) Initial Display for Aircraft. The following illustrates the initial display for an aircraft prior to launch:

> knox1

KNOX1 IS CURRENTLY NOT FLYING - IT IS ASSUMED YOU DESIRE TO LAUNCH.

KNOX1 IS NOW BEING LAUNCHED

MOVEMENT NOT DESIRED ENTER N(ONE) OTHERWISE ENTER SPEED DESTINATION HEX IN THIS FASHION

>12
>1720
ENTER SEARCH MISSIONS AS FOLLOWS:
0  NONE SCHEDULED
1  ACTIVE MISSION
2  PASSIVE MISSION

> 0
YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
NONE SCHEDULED
DESTINATION 1720 SPEED 12
THERE IS NO SEARCH MISSION SCHEDULED

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> Y

ENTER THE NAME OF THE DESIRED PLATFORM

(4) Full Display for Aircraft. The following illustrates the difference between the initial display and one with combat results.

>knox1

KNOX IS LOCATED AT POSITION 520 AT SPEED 0
IT RELEASED A MK 46 AT A SUBMARINE TARGET LOCATED
AT POSITION 518
THE WEAPON
DETONATED (HIT)
IT CONDUCTED AN ACTIVE SEARCH WITH FOLLOWING RESULTS
SUBMARINE CONTACT LOCATION RANGE
518 2
KNOX1 WEAPON SYSTEM(S)
TYPE RANGE
1 MK 46 20

NO COMBAT ENTER N(ONE) OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

>n
MOVEMENT NOT DESIRED ENTER N(ONE) OTHERWISE ENTER
SPEED
DESTINATION HEX IN THIS FASHION

49
ENTER SEARCH MISSIONS AS FOLLOWS:
0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION

> 1
YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
NONE SCHEDULED
SPEED IS ZERO AT LOCATION 520
AN ACTIVE SEARCH MISSION WILL BE CONDUCTED

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> y

The player should also be careful not to use a weapon which is inappropriate for the target; for instance, cruise missiles against a submarine; nor to employ a weapon system beyond its tactical range. Also note that all commands are echoed prior to exiting from that section. If the player has made erroneous entries, he may enter Y(ES) to continue with another platform and reenter the platform's name over again. The only problem is that the vagaries of the programming will not allow the player to delete an employed weapon system. (All else may be changed.)

It should be noted that the coding is self-explanatory and user-friendly. By following the computer prompting, the players' only errors should be tactical!

(5) **Damage suffered.** The following example illustrates how platform self-damage is displayed just prior to accepting orders:

50
BORIS IS LOCATED AT POSITION 1522 SPEED 2

NOTE
DUE TO PREVIOUS AND/OR CURRENT DAMAGE SUFFERED, MAX SPEED IS NOW 0
THE SHIP HAS BEEN HIT 3 TIMES

IT HAS BEEN HIT BY A TORPEDO
IT HAS BEEN HIT BY A NON-TORPEDO WEAPON
IT HAS BEEN HIT BY A TORPEDO

THE BORIS ATTACK ON TARGET LOCATED AT 720
RELEASE OF 21" RESULTED IN A DETONATION (HIT)

IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)
ACTIVE

<table>
<thead>
<tr>
<th>TYPE</th>
<th>LOCATION</th>
<th>METHOD</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

THERE WERE NO DETECTIONS

SINCE THE BORIS HAS BEEN SUNK, NO COMMAND INPUTS CAN BE ACCEPTED
ENTER (Y)ES TO CONTINUE WITH ANOTHER PLATFORM
ENTER (N)O TO RETURN TO DISPLAY SELECTION

When N(O) is entered, control is returned to the selection of display options.

E. GAME END

As previously discussed, there are three possible ways to end the game.

1. Game Turns

When the number of game turns exceeds the number input by the players, the following message is printed:
YOU HAVE JUST COMPLETED GAME TURN 2.
TYPE Y(ES) IF YOU WANT TO CONTINUE THE GAME, OR N(O) TO ALLOW THE GAME TO END.

If "NO" is entered, the game results are displayed.

2. **Disengagement**

When the number of destroyed ships for one player equals or exceeds the number he input for disengagement, the following statement is printed:

THE NUMBER OF BLUE SHIPS DESTROYED IS EQUAL TO OR GREATER THAN THE NUMBER INPUT FOR GAME END. IF YOU WANT TO CONTINUE THE GAME, TYPE Y(ES). TYPE N(O) TO END THE GAME.

If "NO" is entered, the game results are displayed.

3. **Total Destruction**

The total destruction of one player's ship forces will automatically end the game by displaying the final game results.

F. **FINAL RESULTS**

Game end results in the following information being printed:

**BLUE RESULTS:**

<table>
<thead>
<tr>
<th>SHIP</th>
<th>STATUS/HITS</th>
<th>FINAL LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TINCAN</td>
<td>0</td>
<td>1102</td>
</tr>
<tr>
<td>ESCORT</td>
<td>SUNK</td>
<td>720</td>
</tr>
<tr>
<td>LAPON</td>
<td>0</td>
<td>714</td>
</tr>
</tbody>
</table>

TYPE Y(ES) TO SEE RED RESULTS.

>yes
RED RESULTS:

<table>
<thead>
<tr>
<th>SHIP</th>
<th>STATUS/HITS</th>
<th>FINAL LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVAN</td>
<td>SUNK</td>
<td>1311</td>
</tr>
<tr>
<td>BORIS</td>
<td>SUNK</td>
<td>1522</td>
</tr>
</tbody>
</table>

The players are also given the option of starting a new game:

TYPE Y(ES) TO RETURN TO MENU SELECTION AND START A NEW GAME. TYPE N(O) TO EXIT THE GAME.

> y
IV. PROGRAM DOCUMENTATION

A. INTRODUCTION

This chapter contains documentation for the approximately 2350 lines of FORTRAN source code which comprise the computer version of Up Scope. It is intended primarily for the reader who is interested in modifying the program. In using this chapter, the reader should refer to the source listing in Appendix F.

The program can be logically divided into the following parts:

1. Dimension, Common and Data Statements
2. Game Parameters and Weapon Systems Initialization
3. Game Control
4. Display Subroutines
5. Combat Execution Subroutines
6. Utility Subroutines

Each of the six parts will be described individually in the following pages. However, prior to a description of the FORTRAN source program, one should obtain an understanding of the major arrays and variables used in the program.

B. DESCRIPTION OF THE MAJOR ARRAYS

This section will only discuss the major arrays such as those used in the common statement and dimensioned in the main program for data arrays. All arrays are integer.
1. **Arrays in common storage**

   a. **PLATFM(100,37)**

   PLATFM is the largest and most important of all the arrays because it contains the basic information about each selected platform. Each platform has 37 columns associated with it. The size of the row dimension of PLATFM limits the game to 100 weapon systems. Aircraft, surface ships and submarines are included in the array. Due to the difference of their attributes, the columns of PLATFM associated with aircraft do not always serve the same purpose as those associated with ships, and only the first 24 columns of rows dealing with aircraft contain information. Any differences are discussed below.

   (1) **PLATFM(xxx,1).** This column is used as a combat assignment flag. If the platform is an aircraft, a zero indicates no combat is scheduled, a one indicates combat is scheduled and weapon one is to be used and a two indicates weapon two is to be used. If the platform is a ship, a zero indicates no combat and a one indicates combat. This difference is due to the fact that aircraft can only fire one weapon per game turn while ships can fire up to three.

   (2) **PLATFM(xxx,2).** This column holds the combat location which is the hex address to which the weapon/ weapons will be delivered.
(3) **PLATFM(xxx,3)**. This is the hex address of the current location of the platform.

(4) **PLATFM(xxx,4)**. Current speed of the platform.

(5) **PLATFM(xxx,5)**. This is the hex address of the future location of the platform.

(6) **PLATFM(xxx,6)**. This column is used as a disturbed water flag. A zero means the platform is not within one hex of a disturbed water zone and a three means it is. A three is used because the routine that controls detections adds the contents of this column to the ship's evasion value.

(7) **PLATFM(xxx,7)**. Column seven controls the search mission assigned to the platform. A zero indicates no search mission is scheduled, a one means the active sonor is to be used, a two means the passive sonor is to be used and a three means both are to be used.

(8) **PLATFM(xxx,8)**. This column indicates the platform type. A zero means the platform is an aircraft, a one means a surface ship and a two means a submarine.

(9) **PLATFM(xxx,9)**. This column is not used at present and is reserved for possible improvements to Up Scope.

(10) **PLATFM(xxx,10)**. This column is not used at present and is reserved for possible improvements to Up Scope.

(11) **PLATFM(xxx,11)**. For aircraft, this column contains a zero or a one. A zero means the aircraft is not
flying and a one means it is flying. For ships, the column contains the number of durability points lost.

12) PLATFM(xxx,12). Max speed of the platform.

13) PLATFM(xxx,13). Active detection value of the platform.

14) PLATFM(xxx,14). Passive detection value of the platform.

15) PLATFM(xxx,15). The number of type one weapons carried by the platform.

16) PLATFM(xxx,16). The number of type two weapons carried by the platform.

17) PLATFM(xxx,17). For aircraft, this is the row index of the type one weapon. For ships, this column holds the number of type three weapons carried by the platform.

18) PLATFM(xxx,18). For aircraft, this is the row index of the type two weapon. For ships, it holds the evasion value.

19) PLATFM(xxx,19). For aircraft, this column is not used. For ships, it is the defense value of the ship.

20) PLATFM(xxx,20). For aircraft, this value is set at one. For ships, it contains the durability points of the ship. It will become zero when the ship is sunk.

21) PLATFM(xxx,21). For aircraft, this column holds the row index number of the ship to which it belongs. For ships, it holds the index of the type one weapon.
(22) \textit{PLATFM(xxx,22)}. For aircraft, this is the row index for the type of aircraft. For ships, it holds the row index of the type two weapon.

(23) \textit{PLATFM(xxx,23)}. For aircraft, this column holds a number associated with the probability that weapon one hit its target. A zero means the weapon was not fired during the last game turn, a one through six means the number of chances out of six the weapon had of hitting its target, a seven means the weapon fired was a torpedo and there was a detonation and an eight means the weapon was a torpedo and there was no detonation. For ships, it holds the row index of the type three weapon.

(24) \textit{PLATFM(xxx,24)}. For aircraft, this column holds the same information as \textit{PLATFM(xxx,23)} except for weapon two. For ships, it is a flag that indicates if simultaneous active and passive search missions are possible. A zero means simultaneous missions are not possible and a one means they are.

(25) \textit{PLATFM(xxx,25)}. This is a flag that contains a zero if the ship's passive sonar system suffers no degradation from speed and a one if degradation is caused by speed.

(26) \textit{PLATFM(xxx,26)}. The row index of the type one aircraft carried by the ship.

(27) \textit{PLATFM(xxx,27)}. The row index of the type two aircraft carried by the ship.
The number of type one aircraft carried by the ship.

The number of type two aircraft carried by the ship.

The class row index of the ship. These numbers link the platform with the information found in Appendix B.

The PLATFM row index number of the first type one aircraft carried by the ship.

The PLATFM row index number of the first type two aircraft carried by the ship.

The PLATFM row index number of the last type one aircraft carried by the ship.

The PLATFM row index number of the last type two aircraft carried by the ship.

This column holds a number associated with the probability that weapon one hit its target. A zero means the weapon was not fired during the last game turn, a one through six means the number of chances out of six the weapon had of hitting its target, a seven means the weapon fired was a torpedo and there was a detonation and an eight means the weapon was a torpedo and there was no detonation.

The same information as PLATFM(xxx,35) except for weapon two.
(37) PLATFM(xxx,37). The same information as PLATFM(xxx,35) except for weapon three.

b. DETECT(100,3)

The DETECT array is used to store the detections that occur during a game turn. After each turn, the first column is initialized to minus one. If a game turn results in more than 100 detections a message is printed, but the game continues. All detections beyond 100 are lost.

(1) DETECT(xxx,1). Column one contains the PLATFM row index of the platform that is detected.

(2) DETECT(xxx,2). Column two contains the PLATFM row index of the platform that is detected.

(3) DETECT(xxx,3). Column three contains a flag indicating the type of sensor used. A zero means passive and a one means active.

(4) DETECT(xxx,4). Column four contains the distance in hexes between the detected and the detecting platforms.

c. WEP(29,3)

The WEP array contains the basic information about the 29 types of weapons portrayed by the game. The same information is contained in the data array WEPDA. The two are set equal so WEP can be commoned. The information contained in the array can be found in the Weapon Data Summary, Appendix C.
(1) **WEP(xx.1).** This holds the accuracy rating of the weapon.

(2) **WEP(xx.2).** This is the range of the weapon in hexes.

(3) **WEP(xx.3).** This is the index for the type of weapon. A one indicates a missile, a two a missile/depth bomb, a three a torpedo and a four indicates a depth charge.

d. **HIT(100,3)**

   The HIT array is used to maintain the results of combat for a game turn. At the end of each turn the results are entered into PLATFM, and the first column of the HIT array is initialized to minus one.

   (1) **HIT(xxx,1).** The PLATFM row index of the platform that fired the weapon.

   (2) **HIT(xxx,2).** The row index of the type of weapon fired.

   (3) **HIT(xxx,3).** The PLATFM row index of the platform hit by the weapon.

e. **SUNK(100)**

   SUNK is the vector of PLATFM row index numbers of ships that have been sunk. This is a cumulative list in order of the occurrence of each sinking.

f. **WATER(100)**

   WATER is a vector of the hex addresses that contain disturbed water as a result of a weapon detonating.
It is initialized to negative one prior to the combat execution phase.

g. NAME(100,6)

NAME is an alpha array that contains the user-supplied names of platforms. Each row is associated with the same numbered row in PLATFM. Its format is 6A1.

h. BUFFER(100,7)

BUFFER is the array used to hold the combat, move and search commands until the next game turn is started. This allows the user to recall each display option as many times as desired without disturbing the information contained in PLATFM. This information is input to PLATFM when the combat driver CONTRL is called.

(1) BUFFER(xxx,1). This column holds the combat assignment flag associated with PLATFM(xxx,1)

(2) BUFFER(xxx,2). This is the combat location desired for the next game turn.

(3) BUFFER(xxx,3). This is the speed desired for the next game turn.

(4) BUFFER(xxx,4). This is the desired location for the next game turn.

(5) BUFFER(xxx,5). This column assigns the type of search mission for the next game turn.

(6) BUFFER(xxx,6/7/8). These columns are used by non-aircraft to assign which weapons will be employed.

62
i. BLUPAS(6)

BLUPAS is the array that holds the blue player's alpha password. It is used before the initial loading of weapon systems and before each game turn to ensure the players cannot enter instructions to the other's platforms.

j. REDPAS(6)

REDPAS contains the red player's password and is used in the same way as BLUPAS.

3. Data Arrays

The data arrays (most of those used in the dimension statement) contain the basic information about the weapon systems portrayed by the game. Some of them are equivalence to ease their entry and understanding. This will be discussed in the section dealing with the initial portion of the main program.

a. SHIPDA(130,18)

SHIPDA is the data array that holds the basic information about the ships portrayed by Up Scope. This information can be found in Appendix B. This information is transferred to PLATFM during the initial loading of the user selected weapon systems. Each column in SHIPDA is equivalence to one of the arrays S1 through S18, and the data statements are actually in terms of the S arrays.

(1) SHIPDA(xxx,1)/S1(xxx). Max speed of the ship.
(2) \text{SHIPDA}(xxx,2)/S2(xxx). Active detection value of the ship.

(3) \text{SHIPDA}(xxx,3)/S3(xxx). Passive detection value of the ship.

(4) \text{SHIPDA}(xxx,4)/S4(xxx). A flag that is zero if active and passive sonar cannot be used simultaneously and a one if they can.

(5) \text{SHIPDA}(xxx,5)/S5(xxx). A flag that is zero if the passive sonar system capability is degraded by the speed of the platform and a one if no degradation occurs.

(6) \text{SHIPDA}(xxx,6)/S6(xxx). Evasion value of the ship.

(7) \text{SHIPDA}(xxx,7)/S7(xxx). Defense value of the ship.

(8) \text{SHIPDA}(xxx,8)/S8(xxx). The index of the type one weapon carried by the ship.

(9) \text{SHIPDA}(xxx,9)/S9(xxx). The number of type one weapons carried by the ship.

(10) \text{SHIPDA}(xxx,10)/S10(xxx). The index of the type two weapon carried by the ship.

(11) \text{SHIPDA}(xxx,11)/S11(xxx). The number of type two weapons carried by the ship.

(12) \text{SHIPDA}(xxx,12)/S12(xxx). The index of the type three weapon carried by the ship.
(13) \text{SHIPDA}(xxx,13)/S13(xxx). The number of type three weapons carried by the ship.

(14) \text{SHIPDA}(xxx,14)/S14(xxx). The index of the type one aircraft carried by the ship.

(15) \text{SHIPDA}(xxx,15)/S15(xxx). The number of type one aircraft carried by the ship.

(16) \text{SHIPDA}(xxx,16)/S16(xxx). The index of the type two aircraft carried by the ship.

(17) \text{SHIPDA}(xxx,17)/S17(xxx). The number of the type two aircraft carried by the ship.

(18) \text{SHIPDA}(xxx,18)/S18(xxx). Durability value of the ship.

b. \text{NAMESH}(130,8)

\text{NAMESH} is an alpha array that contains the class names of the ships found in \text{SHIPDA} and Appendix B. Each column of \text{NAMESH} is equivalenced to one of the arrays NS1 through NS8, and the data statements are actually in terms of the NS arrays. The format of \text{NAMESH} is 8A2.

c. \text{ACDATA}(16,7)

\text{ACDATA} is the data array that contains the aircraft information. The information it contains can be found in Appendix D.

(1) \text{ACDATA}(xx,1). Max speed of the aircraft.

(2) \text{ACDATA}(xx,2). Active detection value of the aircraft.
(3) **ACDATA(xx,3)**. Passive detection value of the aircraft.

(4) **ACDATA(xx,4)**. The index of the type one weapon carried by the aircraft.

(5) **ACDATA(xx,5)**. The number of type one weapons carried by the aircraft.

(6) **ACDATA(xx,6)**. The index of the type two weapon carried by the aircraft.

(7) **ACDATA(xx,7)**. The number of the type two weapons carried by the aircraft.

d. **NAMEAC(16,6)**

NAMEAC is an alpha array that contains the names of the 16 aircraft. The format of NAMEAC is 6A2.

e. **WEPDA(29,3)**

WEPDA is the data array that contains the weapon data. It is equivalent to WEP which is discussed in the section dealing with the arrays in the common statement.

f. **NAMEWP(29,6)**

NAMEWP is an alpha array that contains the names of the weapons. Its format is 6A2.

g. **INPUT(70)**

This array is used to read in strings of alpha characters. All inputs are read in this way and then converted to the mode and format of the information actually required.
h. ALPHA(2)

ALPHA(1) contains a 'Y' and ALPHA(2) contains an 'N'. This array is used to compare user answers to questions that require a yes or no answer.

i. CHECK(6)

CHECK is used to compare user inputs to the Red or Blue passwords.

j. CLASNA(8)

CLASNA is used to read in the ship class names entered by the user.

C. DESCRIPTION OF MAJOR VARIABLES

The variables described in this section include only the major variables like those found in the COMMON statement. Other variables will be discussed as they appear in the program description. Normally single letter variables are used as local and temporary variables.

1. IPLAT

This variable is the total number of platforms loaded in the PLATFM array. It is used for loops that act on aircraft, surface ships and submarines.

2. IAIR

IAIR is the total number of aircraft loaded in the PLATFM array.

3. ISURF

ISURF is the total number of surface ships loaded in the PLATFM array.
4. **ISUB**
   ISUB is the total number of submarines loaded in the PLATFM array.

5. **ADV**
   ADV is the variable used to represent a ship's active detection value. It is used between all the detection subroutines.

6. **PDV**
   PDV is the same as ADV for passive detection value.

7. **AEV**
   AEV is similar to ADV and PDV except it represents a ship's active evasion value.

8. **PEV**
   PEV is used the same way as AEV except for passive evasion values.

9. **ERATIC**
   This variable is a flag that indicates the water condition of the playing area.

10. **BLUAIR/REDAIR**
    These variables hold the number of Blue/Red aircraft entered by the players.

11. **BLUSUR/REDSUR**
    These variables hold the number of Blue/Red surface ships entered by the players.

12. **BLUSUB/REDSUB**
    The number of Blue/Red submarines entered.
13. **PBAIR1/PBAIR2**

All the variables of this form are pointers to the PLATFM array, and they are actually acronyms for what they represent. P(latform)B(lue)AIR1 is the row index number of the first Blue aircraft entered in PLATFM. P(latform)B(lue)AIR2 is the row index number of the last Blue aircraft entered in PLATFM. These variables are used in the initial loading of the PLATFM array and in the display subroutines. Figure 4 shows how the pointers are aligned, and how the data in PLATFM is arranged.

14. **PRAIR1/PRAIR2**

These variables are pointers to the first and last Red aircraft in PLATFM.

15. **PBSUR1/PBSUR2**

Pointers to the first and last Blue surface ships in PLATFM.

16. **PRSUR1/PRSUR2**

Pointers to the first and last Red surface ships in PLATFM.

17. **PBSUB1/PBSUB2**

Pointers to the first and last Blue submarines in PLATFM.

18. **PRSUB1/PRSUB2**

Pointers to the first and last Red submarines in PLATFM.
Figure 4. Format of the PLATFM Array
19. TURN

TURN stores the total number of game turns the players choose to play.

D. DIMENSION, COMMON AND DATA STATEMENTS

The first 194 statements of the program are dedicated to dimensioning the major arrays and setting up the data base required for Up Scope.

1. Line 1

All variables and arrays used in the program are integer except for: R which is used in the shot result subroutine to express the probability of a weapon hitting a target, RN which is used in FIRE to generate a random number and Y which is used in AEVSUB and PEVSUB in calculating evasion values.

2. Lines 2 through 17

The DIMENSION statement includes all the arrays discussed in section B.2. of this chapter. The majority of these arrays are used to ease the entry of data and are equivalenced to the arrays actually used by the program. The use of so many arrays eased the data entry because each array of the Sx type contained only one type of ship information. For example, S6 contains all the evasion values for the 130 ship types in Up Scope. If the evasion value of the America was wrong, the error would be in the second element of S6. It is in the second element because
the America is the second ship type in the data contained in Appendix B.

3. Lines 18 through 25
   All the arrays and variables in the COMMON statement were discussed earlier in this chapter.

4. Lines 26 through 191
   These lines are the actual data statements used to load the basic platform and weapon information. Each one of these arrays was discussed in section B.2. of this chapter. As previously mentioned, each Sx array is one column of the SHIPDA array, and each NSx array is one column of the NAMESH array.

5. Line 192
   MAXPLT is used as the maximum number of platforms that can be played by the game. This limit is imposed by the size of the PLATFM and NAME arrays.

6. Line 193
   ALPHA is an array used to check the player's answers to "yes" and "no" type questions.

E. GAME PARAMETERS AND WEAPON SYSTEMS INITIATION
   Lines 194 through 742 are used to initialize all the arrays and variables, load the basic game parameters and to load the player-selected platforms.

1. Lines 194 through 201
   This is a loop that sets the array WEP equal to the data array WEPDA so WEP can be placed in the COMMON statement.
2. **Lines 202 through 211**
   These lines initialize the game parameter variables to zero.

3. **Lines 212 through 231**
   These lines contain a nested loop that sets PLATFM, WATER, DETECT, HIT, SUNK and BUFFER to their initial values.

4. **Lines 232 through 245**
   These lines print the format statement that gives the menu of choices and reads the player's choice. The alpha array INPUT is used to read the number of the player's choice as a character string, and the subroutine NONVRT discards any leading blanks and returns the integer form of the choice.

5. **Lines 246 through 256**
   This group of statements will cause new arrays entered by the players to be reinitialized. It is an option that will allow the players to start a new game. Once the option is selected the players are asked if they want to continue and are given a chance to return to menu selection without erasing the arrays. The players answer to this question is read by INPUT, and any leading blanks and characters other than the first one are discarded by INVRT1. If the players want to restart the game, control is transferred to 99(line 205).
6. Line 257
   This line transfers control to the STOP statement if the players indicate a desire to stop play and exit the game.

7. Lines 258 through 284
   These statements print a warning about the maximum number of platforms allowed by the game, and read the number of game turns the game is to last and the water condition.

8. Lines 285 through 324
   These lines read the basic game parameters and the password for the Blue side. The password is read with INPUT, and INVRT6 discards any leading blanks and any characters other than the first six. It returns the password as BLUPAS which is echo-printed to ensure the player knows his password as “understood” by the program. The variables BLUSUR, BLUSUB and BLUDES are also input and echoed by these lines with an opportunity to correct any errors.

9. Lines 325 through 364
   Lines 325 through 364 perform for the Red player the same function lines 285 through 324 do for the Blue player.

10. Lines 365 through 375
    These statements assign numbers to variables used as pointers to the PLATFM array during the initial loading of the platforms. Because the program automatically loads the aircraft carried by a ship and the combat, move and
search routines require aircraft to appear first in the array, surface ships and submarines are located at the bottom of the PLATFM array using these pointers. This method of loading leaves a gap between aircraft and ships unless there are exactly 100 platforms loaded. The gap, if it exists, is closed prior to the start of the game. Figure 4 as previously shown, shows the form of the PLATFM array and its pointers prior to closing the gap.

11. **Lines 376 through 534**

This section of the program performs the loading of the platforms selected by the Blue player.

a. **Lines 379 through 397**

These lines ask for, and read, the Blue player's password. If the reply is incorrect, he is given the opportunity to reenter the password or return to menu selection. DO LOOP 141 compares each element of the array CHECK with BLUPAS. A is set equal to one if any elements do not match and an error message is printed. If the entered password is correct, control is transferred to 8 (line 398) and the player proceeds with loading.

b. **Lines 398 through 446**

This section controls the loading of the Blue surface ships. SHIPNO is a local variable that is used to store the number of the Blue ships being loaded. It runs from zero to BLUSUR. Prior to entry into the loading loop, BLUSUR is checked against zero, and control is transferred
to the submarine loading section if no Blue surface ships were desired by the player. DO LOOP 165 is a long loop that does all the surface ship loading. It runs from PBSUR1 to PBSUR2 using SHIP as its index. The first steps of the loop are used to increment SHIPNO by one and print a format statement requesting a six character name for the SHIPNO ship. This player assigned name is stored in NAME(SHIP,x). This player assigned name is stored in NAME(SHIP,x). SHIP is also the row index number of PLATFM containing the information associated with NAME(SHIP,x). Next the player is asked to enter the ship class name of SHIP. The proper class names and the specifications for each class can be found in Appendix B. CLASNA is used to store the class name, and subroutine SCLASS is called to find the row index number of the class name in NAMESH that matches CLASNA. This index is stored in CLASIN, and it is associated with the row in SHIPDA that contains the basic information for that class of ship. CLASIN is stored in PLATFM(SHIP,30). The next piece of information entered is the initial hex location for the ship which is stored in PLATFM(SHIP,3). The last 19 statements of this section transfer the ship class information from SHIPDA and enter it in the appropriate columns of PLATFM. Sections B.1. and B.2. of this chapter discuss the contents of the columns of these arrays.
c. Lines 447 through 485

DO LOOP 155 is the loop that loads any aircraft SHIP carries. It runs from 26 to 27 because these are the columns of PLATFM that contain the aircraft index numbers. The first statement of the loop checks the column to see if an index number is entered. If the number is zero, control passes to the end of the loop. If the location contains an aircraft index number, the number is stored in ACINDX. The next statement prints the class name of the ship being loaded, the number of ACINDX type aircraft the ship carries and the name of the ACINDX type aircraft as stored in NAMEAC. Line 456 stores the row index number of the first ACINDX type aircraft in the parent ship's PLATFM row. Line 457 adds the number of ACINDX type aircraft carried by the ship to BLUAIR, and line 458 stores the row index number of the last ACINDX type aircraft in the parent ship's PLATFM row. DO LOOP 154 actually loads the aircraft information into PLATFM. Its index is AC, and it runs from P to Q which are set equal to the first and last PLATFM row index numbers of the aircraft. ACOUNT is the number of the aircraft being loaded. DO LOOP 154 also asks the player for a six character name for each aircraft. DO LOOP 153 enters the name in the NAME array. The next 12 lines transfer the required information from ACDATA to PLATFM.
d. Lines 486 through 534

This section loads the Blue submarines. The first statement checks BLUSUB to see if the Blue player entered a request for submarines. If not, control is transferred to the first statement of the Red loading section. DO LOOP 175 controls the loading, and it runs from PBSUB1 to PBSUB2. SUB is the index of the loop and SUBNO is the number of the submarine being loaded. The submarine loading statements follow the same pattern as the surface ship section except no aircraft are loaded. A six character name for the submarine is requested and entered in the NAME array. A class name and an initial hex location are also requested. The last 18 lines transfer data from SHIPDA to PLATFM.

12. Lines 535 through 692

These statements perform the loading of Red platforms, and they are almost an exact duplication of the Blue loading section. The only difference is in line 57 which sets REDAIR equal to BLUAIR. This is necessary to ensure the Red aircraft are properly loaded into PLATFM using the incremental process starting in line 615.

13. Lines 693 through 710

The first five lines of this section set the aircraft PLATFM pointers. The variables for each type of platform and total number of platforms are also set. The total number of platforms, IPLAT, is compared to 100. If it
is larger, the game must be restarted, and control is
transferred to 99 (line 205).

14. Lines 711 through 742

These lines close the gap between aircraft and
ships loaded into PLATFM. The first statement checks the
location of the first Blue surface ship, PBSUR1, against
the row index number of the last Red aircraft, PRAIR2, plus
one. If these numbers are equal, there is no gap and the
section is passed. If there is a gap, it is closed with a
group of nexted do loops. DO LOOP 283 runs from one to
ISHIP, the total number of Blue and Red ships, using I as
an index. This loop moves through all rows of the PLATFM
array that contain ships. DO LOOP 281 actually moves the
information for each column of the selected rows. DO LOOP
282 closes the gap in the NAME array. DO LOOP’s 284 and
285 check the ship’s PLATFM row to see if it contains
aircraft. If aircraft have been assigned, the PLATFM rows
containing aircraft have new parent ship row index numbers
placed in column 21 to adjust for closing the gap.

F. GAME CONTROL

Lines 743 through 877 of the main program control the
actual "play" of the game turns. This is accomplished by
several administrative functions and calls to subroutines.
A simplified flow chart can be found in Figure 5. ITURN
is the counter for the number of game turns played, and it
Figure 5. Simplified Game Flowchart
Figure 5 (continued)
is compared to the number stored in TURN to determine the normal game end.

1. **Lines 747 through 759**

   The first action required for game control is to gain access to the display routines. This is done by requesting the Blue player to input his password and by calling subroutine STATUS discussed later in this chapter. STATUS will check the password and return an error code of zero in ERROR if the password is correct. If ERROR equals zero, STATUS will allow the player to view the results of the previous game turn and input instructions for the next turn. Control will then be transferred to 20 (line 760), the Red call to STATUS. If ERROR equals one, an error message is printed, and instructions are requested. The player can either reenter his password or return to menu selection to restart the game.

2. **Lines 760 through 770**

   These lines perform the same function for the Red player as the previous section did for the Blue player.

3. **Line 771**

   Line 771 calls subroutine CONTRL, discussed later in this chapter, which is the driver for all the combat, move and search routines.

4. **Lines 772 through 781**

   BDES and RDES are local variables used to store the number of Blue and Red ships sunk after each game turn.
LOOP 303 loops through array SUNK, and increments the appropriate variable.

5. **Lines 782 through 793**

   This section of statements asks several questions about BDES. If BDES is equal to the total number of Blue ships, the game is automatically ended by transferring control to line 815. If BDES is less than BLUDES, the player-selected number for premature game end, control is passed to 23 (line 792) which checks RDES. The last eight lines are reached only if BDES is less than the total number of Blue ships and greater than or equal to BLUDES. They print a statement informing the player of the situation, and give him the option of continuing or ending the game.

6. **Lines 792 through 801**

   These lines check RDES for the Red player as lines 782 through 793 do for the Blue player.

7. **Lines 802 through 811**

   Lines 802 through 811 increment ITURN and transfer control to the start of another game turn if it is less than TURN. If it is greater than or equal to TURN, the players are given the option of continuing or ending the game.

8. **Lines 812 through 878**

   This section of statements ends the game by printing the status and final location of each ship. COUNT is a counter used to ensure the results are not pushed beyond the top of a CRT display. DO LOOP 313 prints the results of
the game. It runs from A to B with PLT as its index. A and B are variables set equal to the first and last PLATFM row index numbers of each of the four types of ships. C is a flag that indicates which of the four is being printed (one-Blue surface, two-Blue submarines, three-Red surface and four-Red submarines). Lines 833 and 834 increment COUNT and compare it to ten. If COUNT is less than ten, control is transferred to the CONTINUE statement of the loop. If COUNT is equal to ten, a new page of output is started. Lines 843 through 845 check the value of C, and transfer control to the appropriate section to increment A, B and C. For example, if C is equal to one, control passes all three statements, and goes to the Blue submarine section. There A is set equal to PBSUB1, B to PBSUB2 and C is set equal to two. Control then returns to the loop to print the Blue submarine results. If C is equal to two, control is transferred to line 863 which prints a message to begin the Red results. Then A, B, C and COUNT are updated for Red surface ships, and control is again transferred to the loop. After all four ship results have been printed, line 868 prints a message that gives the players a choice of starting a new game by returning to menu selection or letting the game end. If the game is allowed to end, line 874 prints an exit message.

G. DISPLAY SUBROUTINES

The display subroutines are called every game turn. They are used to display the results of the game turn and to
input combat, move and search instructions for the next turn. Four display options are available. They are detections of enemy ships, results of weapons fired, damage to friendly ships and individual display/input instructions.

1. Subroutine \texttt{STATUS(PASWRD,ERROR,NAMEWP)}

\texttt{STATUS} is the main driver for all the display routines. It is called with the player's password, \texttt{PASWRD}, and the array \texttt{NAMEWP}, and it returns an error flag in \texttt{ERROR} if the player entered the wrong password. No modification for Blue or Red calls is required because the subroutine assigns appropriate local variables based on the entered password.

a. Lines 13 through 30

\texttt{DO LOOP 100} checks \texttt{PASWRD} against \texttt{BLUPAS}, and sets \texttt{A} equal to one if one of the elements does not match. If this is the case, control is transferred to 300 (line 31). If \texttt{PASWRD} does match, the calling player is Blue and lines 19 through 29 set local variables equal to the Blue \texttt{PLATFM} pointers and Blue game parameters.

b. Lines 31 through 47

These lines check \texttt{PASWRD} against \texttt{REDPAS} and if there is a match, perform the same operations as lines 14 through 29.
c. Lines 48 through 51

If neither BLUPAS or REDPAS match PASWRD, the player has made a mistake. These lines print an error message, and transfer control to the end of the subroutine with ERROR equal to one.

d. Lines 52 through 64

If either BLUPAS or REDPAS match PASWRD, control is passed to this section. The first seven lines print a format statement that asks the player to select and enter a display option. The answer is read by INPUT, converted to an integer by NONVRT and stored in A. Line 66 compares A to the possible option numbers, and prints an error message if it is incorrect. If A is a proper option, control is transferred to 830 (line 71).

e. Lines 71 through 82

These lines check A, and call the various display subroutines that correspond to the options. After each call, control is returned to 1000 (line 59), and the option menu is reprinted.

f. Lines 83 through 92

The last option is to start the next game turn. These lines print a format statement that asks the player if he has entered all combat, move and search instructions. If he does not answer yes, control is passed to line 59. Otherwise control is returned to the main program.
2. **Subroutine DETXT(AC1,AC2,SUR1,SUR2,SUB1,SUB2,NAMEWP)**

This subroutine loops through the DETECT array and prints the detections of the calling player's platforms. ALPHA1 is an alpha array that contains the word "SURFACE" in its first row and "SUBMARINE" in the second. ALPHA2 is an array that contains the words "PASSIVE" and "ACTIVE". COUNT is a counter for the number of lines the routine prints. When COUNT goes past a certain number, a new "page" of output is stated. This prevents detections from being pushed beyond the top of CRT display units.

DO LOOP 6 is the only loop in the subroutine, and it runs from one to 100 with PLT as its index. PLT is the number of a row of the DETECT array. D1 is set equal to the first column of DETECT which contains the PLATFM row index number of the platform doing the detecting. D2 is set equal to column two of DETECT which contains the row index number of the detected ship. If the first column of a row in the DETECT array is minus one, the previous row was the last detection, and control is transferred to the end of the routine. Lines 22 and 23 check the detecting platform to ensure it belongs to the calling player. If it does not, that row of the array is passed. If the detecting platform does belong to the calling side, line 25 prints four pieces of information. The first is the name of the detecting platform as stored in the D1 row of NAME. The second is the type of sonor used. A flag relating to the sonor type is
stored in the third column of DETECT. A one is added to this flag (zero-passive, one-active), and that row of ALPHA2 is printed. The third piece of information is the type of ship detected which is an index number stored in column eight of row D2 in PLATFM. This number yields the row of ALPHA1 which contains the ship type name. The last piece of information is the hex location of the D2 ship in PLATFM. Lines 28 and 29 increment COUNT and check it against the number 19. If COUNT is greater than or equal to 19, COUNT is set to zero, the heading is printed on a new page and control is passed back to DO LOOP 6. Otherwise control is maintained by the loop.

3. Subroutine SHOTRE(AC1,AC2,SUR1,SUR2,SUB1,SUB2, NAMEWPSUR,SUR,SUB)

SHOTRE is the subroutine that displays the results of the weapons fired by the calling side. R is a real variable that stores the probability of a hit, and ALPHA1 is an array that holds the word "DETONATION" in row one and "NO DETONATION" in row two. COUNT is again used to count the number of lines printed so the results are not pushed beyond the top of a CRT display.

a. Lines 16 through 46

These lines print the results of weapons fired by aircraft. Line 16 transfers control to the ship loop if no aircraft are loaded. DO LOOP 10 runs from AC1 to AC2 with PLT as its index. Each number that PLT takes will be a
row index number of an aircraft in PLATFM of the calling side. Columns 23 and 24 of PLATFM rows containing aircraft hold the shot results for weapons one and two respectively. If both these columns are zero, neither weapon was fired, and line 21 will transfer control to the end of the loop. If one of the aircraft's weapons was fired, line 22 prints the name of the aircraft, the name of its parent ship and the combat location. DO LOOP 7 loops through both weapons carried by the aircraft to see which one was fired. Only one weapon can be fired by an aircraft per game turn. RESULT is set equal to the column which holds the results of the weapon firing. The numbers one through six represent the number of chances out of six that the weapon scored a hit. The number seven or eight is used when the weapon fired was a torpedo. A seven indicates the torpedo detonated (a hit) and an eight means it did not detonate (a miss). Line 30 prints the number of the weapon, the name of the weapon as stored in NAMEWP and the probability of a hit if the shot result is between one and six. If the result is seven or eight, line 34 prints the number and name of the weapon and "DETONATION" or "NO DETONATION". COUNT is then incremented and checked against five. If COUNT is greater than or equal to five, a new page of output is started and COUNT is set equal to zero.
b. Lines 47 through 69

These lines print the weapon result for ships, starting with the surface ships. A is set equal to SUR1, and B is set equal to SUR2. These are the PLATFM row index numbers of the first and last surface ships of the calling side. C is set equal to one as an indication that surface ships are being processed. DO LOOP 16 runs from A to B with PLT as its index. Line 57 checks the column that holds the shot results for ships, and passes control to the end of the loop if none of the weapons for the PLT ship have been fired. Next the name of the ship and its combat location are printed. DO LOOP 15 runs from weapon one to weapon three for each ship using WP as its index. It checks the results of the WP weapon, and prints a statement giving the probability of hit if RESULT is between one and six or a detonation result.

c. Lines 70 through 75

These lines increment COUNT, compare it to four and start a new page of output if COUNT is equal to four.

d. Lines 76 through 81

Line 76 checks the value of C, and passes control to the RETURN statement if it is greater than one. If C is equal to one, DO LOOP 16 has just processed the surface ship results, and must now do the submarines. C is set equal to two, A and B are set equal to SUB1 and SUB2, and control is transferred to the loop.
4. **Subroutine DAMRE(SUR1,SUR2,SUB1,SUB2,SUR,SUB)**

   DAMRE prints the damage to the calling side's ships using the same double loop approach as SHOTRE.

   a. Lines 18 through 32

      DO LOOP 7 loops through the ships in PLATFM, and prints a "SUNK" statement if column 20 is equal to zero. If column 20 is not equal to zero, the ship is not sunk, and column 11 holds the number of hits. This number is then printed. Next COUNT is incremented, and a new page of output is started if it exceeds nine.

   b. Lines 33 through 38

      These lines check C, reset the variables A and B to SUB1 and SUB2 and return control to DO LOOP 7 if it is equal to one. Otherwise control is returned to STATUS.

5. **Subroutine STATIN(AC1,AC2,SUR1,SUR2,SUB1,SUB2,NAMEWP)**

   This subroutine controls the individual platform displays/command acceptance subroutines which are divided between aircraft and surface ship/submarine sections. It is called by subroutine STATUS and calls STATAC, STATSS, NAMEIN, INVERT1 and INVERT6.

   a. Lines 18 through 30

      This section of code prints the heading telling the player that he is in the individual platform status section and to enter the name of the desired platform. It utilizes INVERT6 to preprocess the entry, then calls NAMEIN to convert the name of the desired platform to a row index.
If the name is either incorrect (spelling or otherwise) or belongs to the opposing player, a zero index is returned. If the index returned is zero the program prompts the player to reenter a valid name.

b. Lines 31 through 38

This section of code determines if the row index of the desired platform is an aircraft or a surface ship/submarine and passes control to either STATAC or STATSS respectively. Again, there is a check against the other players platform being processed.

c. Lines 39 through 47

This section of code lets the player continue with another platform selection or return to the status menu selection.

6. Subroutine STATAC(USED, INDEX, INAME, NAMEWP)

This subroutine is all input/output related and displays a selected aircraft's individual status to include results of assigned combat missions, current speed, current location, results of search missions and damage inflicted on the aircraft. It accepts and processes all game turn command inputs for aircraft. USED is utilized by subroutine STATIN and is a flag set to indicate that STATAC has executed. INDEX is the PLATFM array row index of the desired aircraft. INAME is an array that contains the selected aircraft's name. NAMEWP is the array that contains all weapon system names.
Subroutine STATAC is called by subroutine STATIN and calls INVRT1, INVRT6 and DISTAN.

a. Lines 16 through 19
This section of code determines if the aircraft is currently flying. If not, control is transferred to 35 (line 99), thus bypassing prior game turn status.

b. Lines 25 and 26
Displays current location and speed.

c. Lines 27 through 39
This section of code displays results of combat action undertaken. It displays the target location and weapon employed. If weapon system one (two) was used then PLATFM column 23(24) will contain one to eight. If it is one thru six, then a non-torpedo weapon has detonated and the probability of a hit is that number out of six. A seven indicates a torpedo detonated while a eight indicates the torpedo missed and; therefore, did not detonate.

d. Lines 40 through 62
This section of code displays the search mission designated and the DO 20 loop identifies and displays all detections by the designated aircraft.

e. Lines 63 through 71
This section of code utilizes the weapon row index in PLATFM to display the available weapon system(s) name and maximum range.
This section of code accepts and processes an entered target location and weapon system number (one or two). This data is entered in the appropriate section of the BUFFER array. It then transfers control to 40 (line 107) bypassing the section that launches a non-flying aircraft.

Launches a non-flying aircraft by setting column 11 of PLATFM equal to one.

This section of code accepts and processes entered speed and future location inputs. This data is entered in the appropriate section of BUFFER.

This section of code calls DISTAN to determine the distance between future location and current location. It then checks to insure current maximum speed is not exceeded and that the distance computed is less than, or equal to, the speed command. If in error, control is transferred to 40 (line 107) and prompts the player to reenter speed and future location.

This section of code accepts and processes the search command. It insures that the aircraft has a speed command of zero. If not, it transfers to 50 (line 134) and the player restarts this section.
k. Lines 149 through 166. This section of code displays all entered and processed combat, move and search commands.

7. Subroutine STATSS(USED, INDEX, INAME, NAMEWP)

Subroutine STATSS is all input/output related and displays a selected surface ship/submarine individual status; to include results of assigned combat missions, current speed, current location, results of search missions and damage inflicted on the platform. It accepts and processes all game turn command inputs for surface ship/submarines. USED (line 18) is utilized by subroutine STATIN as a flag to indicate that STATSS has executed. INDEX is the PLATFM row index of the desired platform. INAME is an array that contains the selected platform's name. NAMEWP is the array that contains all weapon system names.

a. Lines 24 through 29

This section of code displays the platforms current position and speed, determines from PLATFM(x,11) how many times the platform has been hit and displays the number of hits received during the game.

b. Lines 35 through 42

This section of code searches the HIT array and displays hits (including type weapon), that the platform received during the previous game turn.

c. Lines 48 through 62

This section of code displays results of combat operations undertaken. It displays the target location,
ETHODOLOGY FOR THE CONVERSION OF TACTICAL BOARD GAMES TO COM-ETC(U)

SEP 80  W C GRIGGS, J A THOMPSON

F/0 9/2

UNCLASSIFIED

2 of 3
then the DO 20 loop determines the weapon system(s) employed and displays employed weapon name and results of that attack in the same fashion that section IV.F.6.c does.

d. Lines 68 through 100

This section of code displays the results of the search mission(s). The DO 40 loop processes the DETECT array to display type platform, location, method and target range.

e. Lines 106 through 118

This section of code first determines if the platform has been sunk (during the previous game turn). If so, the subroutine bypasses the command input section and returns. The DO 51 loop displays weapon system(s) name(s) and range(s) available to the platform.

f. Lines 124 through 142

This section of code accepts and processes an entered target location and weapon system number for combat operations. This data is entered in the appropriate section of BUFFER. Column one of BUFFER is set to one if there is a combat mission assigned. Then column six, seven and eight are set to negative one depending on whether or not weapon system number one, two or three is employed. The program then determines if the player entered another weapon system number. If so, it transfers control to 60 (line 133) to accept another weapon system number.
g. Lines 148 through 169

This section of code accepts and processes entered speed and future location inputs for movement. It then calls a utility subroutine, DISTAN, to determine the distance between future location and current location and checks to insure current maximum speed is not exceeded and that the distance computer is less than, or equal to, the speed command. If incorrect, the program transfers to 70 (line 148) and restarts this section.

h. Lines 175 through 185

This section of code accepts and processes the search command. If both active and passive search is designated (three) and the flag to allow that is not set (PLATFM(x,24) equaling one) then control is transferred to 80 (line 175) to restart this section. The search command is then placed in the BUFFER array.

i. Lines 191 through 214

This section of code displays all entered and accepted combat, move and search commands.

H. COMBAT EXECUTION SUBROUTINES

The subroutines described in this section control and execute one complete game turn after the players have entered all their commands. The control, execution flow and layering of subroutines in this section are as follows:
CONTRL

COMBAT
FIRE
DISTUR
RANGE
FILHIT
DAMREC

MOVE
SEARCH
ADVSUB
AEVSUB
PDVSUB
PEVSUB
DETEK

1. Subroutine CONTRL

Subroutine CONTRL controls the execution of combat, move and search phases for one game turn. It initializes all working arrays at the appropriate times. Subroutine CONTRL is called by the main program when all data display, command entry is complete and the players decide to execute the current game turn. It calls subroutines COMBAT, MOVE, SEARCH and utility subroutine DISTAN.
a. Lines 18 through 36

This section of code initializes the disturbed water array WATER and the hit array HIT in the DO 10 loop. In the DO 20 loop it removes previous combat operations and disturbed water zone entries. The DO 21 loop is executed for aircraft only and removes combat result information from PLATFM. The DO 22 loop removes combat result information from PLATFM for surface ships and submarines.

b. Lines 42 through 46

This section of code checks for non-flying aircraft and, for any, takes the future location of the parent ship from the BUFFER array and puts that in the aircraft's future location in BUFFER.

c. Lines 52 through 66

The DO 25 loop overlays the array BUFFER over selected columns of PLATFM. BUFFER contains the individual combat assignments, movement commands and search commands for the current game turn. Aircraft only carry one weapon system and therefore have no need to utilize columns six, seven and eight of BUFFER which activates the individual surface ship and submarine weapon systems.

d. Lines 68 and 69

Executes all combat and movement

e. Lines 75 through 78

The DO 30 loop initializes the detect array.

This must be done after executing COMBAT and MOVE but before
executing SEARCH as SEARCH makes entries in DETECT.
Subroutine COMBAT utilizes those entries.

f. Lines 83 through 95

This section of code compares all current platform locations with the locations of disturbed water zones in the WATER array. The DO 40 loop sets NUM equal to the number of disturbed water locations. The DO 70 loop increments through all platforms bypassing ones that are either sunk or have had a disturbed water zone recorded previously in the current game turn. The DO 60 loop determines the distance from a platform to every disturbed water location and, if the distance is zero or one, places a disturbed water marker, a three, in column six of the array PLATFM.

g. Line 101

Executes the search phase by calling SEARCH.

h. Lines 107 through 119

This section of code reinitializes the BUFFER array prior to exiting the execution stage of the game turn and prior to accepting future entries. BUFFER(x,4), future location, is set equal to the current position in case a player does not issue a move command for that platform.

2. Subroutine COMBAT

This subroutine executes all phases of air/surface to subsurface, and subsurface to surface/subsurface combat for all platforms. It is called by CONTRL and calls FIRE, FILHIT and DAMREC.
a. Lines 14 through 17

The DO 30 loop extends to line 48 and executes the basic combat assignments for all platforms excluding ones that are sunk, and ones that do not have combat assignments. Line 17 directs control to either aircraft or surface ship/submarine combat execution.

b. Lines 23 through 31

This section of code executes a combat assignment for an aircraft. WEPIND is the column in PLATFM that contains the weapon system row index (for the weapon data array WEP) of the weapon system to be employed. WEPSUP is the column in PLATFM that contains the weapon resupply value for the employed weapon system. Subroutine FIRE is called, passing ATTK(row index of attacking platform), WEPIND, and WEPSUP. IHIT (zero or one, a one if the attack resulted in a hit), ITAR (row index of the target platform) and PRHIT (probability of obtaining a hit) is returned by FIRE. PRHIT is expressed as an integer between one and six. X is the weapon row index. Lines 27, 28 and 29 place the combat results into column 23 or 24. If the attack was successful, subroutine FILHIT is called passing the attacking platforms row index, weapon system row index and target platforms row index in that order. FILHIT records the hit. It then transfers control to the damage assessment section.

c. Lines 37 through 48

This section of code executes a combat assignment for a surface ship/submarine. WEPIND and WEPSUP are as
described above in the aircraft section. The DO 20 loop searches the weapon system row index columns looking for a negative row index which signifies a combat mission for that weapon system. When encountered, the negative sign is removed and subroutine FIRE is called. As previously stated above, ATTK, WEPIND, and WEPSUP are passed and IHIT, ITAR and PRHIT are returned. The combat results are entered in the PLATFM array and subroutine FILHIT is called if the attack was successful.

d. Lines 54 through 59
The DO 40 loop calls subroutine DAMREC to process the damage resulting from the game turn. It passes the row index of the hit platform.

e. Lines 66 through 79
The disturbed water array WATER contains the locations of all detonations that occurred during the current game turn and may contain multiple locations from multiple attacks on a target. This section of code removes all duplicate locations and closes up the array WATER.

3. Subroutine FIRE(ATTK,WEPIND,WPLEFT,IHIT,ITAR,PRHIT)
Subroutine FIRE executes a weapon firing, computes weapon and target range and calculates a random number to compare against a computed probability of a hit to determine hit or miss. It also determines disturbed water locations. ATTK is the attacking platform's row index. WEPIND is the employed weapon row index. WPLEFT is the weapon resupply
value. IHIT is one or zero, a one signifying a hit was obtained. ITAR is the targets row index. PRHIT is the computed probability of a hit. Subroutine FIRE is called by subroutine COMBAT and calls subroutine DISTUR, RANGE, DISTAN and RANDOM.

a. Line 18

Immediately returns to COMBAT if the associated weapon resupply is zero.

b. Lines 24 through 38

This section of code decrements the associated weapon resupply by one. The DO 10 loop sets ITAR equal to the row index of a platform whose current location matches the attackers combat location. If there is not a match, the subroutine returns.

c. Lines 37 and 38

Subroutine DISTUR is called to place a disturbed water marker into WATER using the target's future location.

d. Lines 44 through 49

The DO 15 loop sets Y equal to one if the attacker previously detected the target. If not, the subroutine returns.

e. Lines 55 through 59

This section of code insures that only a weapon capable of attacking that particular type of target is employed, if not, the subroutine returns. It then checks to insure that air/surface platforms only attack subsurface
platforms and that subsurface platforms only attack surface or subsurface platforms, otherwise the subroutine returns.

f. Lines 68 through 75

This section of code determines target range (DIST) from DISTAN, weapon accuracy (ACC) and weapon range (RANG) from RANGE. If target range exceeds weapon range, control is transferred to 40 (line 84) to inform the players of that fact. It then calculates the probability of a hit by subtracting the targets defense value (PLATFM(ITAR,19)) from the weapon accuracy (ACC). By definition, if ACC equals three, the probability of a hit is one out of six. The program calls RANDOM and generates an integer random number from one to six. If PRHIT is greater than or equal to the random number, a hit occurs (IHIT equal to one).

g. Line 82

DISTUR is called to place a disturbed water marker at the target's future location if the weapon was a torpedo and a hit.

4. Subroutine DISTUR(LOCAT)

Subroutine DISTUR utilizes LOCAT which is the hex address of a weapon that has detonated and caused a disturbed water zone. It files it in the disturbed water array WATER. It is called by subroutine FIRE and does not call any subroutines.
a. Lines 13 through 15
This section of code finds the next unused row of the array WATER.

b. Lines 16 through 18
This section of code tells the player there has been over 100 detonations, therefore the players cannot be certain which detonations will cause disturbed water zones. It then returns to subroutine FIRE.

c. Line 19
If there is room in WATER, the location of the disturbed water is entered.

5. Subroutine RANGE(INDEX,ACC,RANG)
Subroutine RANGE utilizes weapon system row index (INDEX) to enter the Weapon Data Summary array (WEP) and pass back the weapon system accuracy (ACC) and the weapon system maximum range (RANG). Subroutine RANGE is called by subroutine FIRE and does not call any subroutines.

6. Subroutine FILHIT(ATTKWEPONITAR)
Subroutine FILHIT files all 'hits' on surface ships and submarines as they occur. FILHIT is called by the subroutine COMBAT and does not call any subroutines.

a. Lines 13 through 15
This section of code locates the next unused row of array HIT.
b. Lines 16 through 18

This section of code tells the players when over 100 hits have occurred. It should be noted that if this happens, only the weapon result displays to the players are affected and that weapon results for some firing will not be available. The actual damage assessment and capability reduction is conducted regardless.

c. Lines 19 through 21

This section of code enters the target's row index, weapon system row index and attacking platform's row index in that order.

7. Subroutine DAMREC(ITAR)

Subroutine DAMREC processes damage inflicted by a hit into the PLATFM array, and places ships that are sunk in the SUNK array. It is called by subroutine COMBAT and does not call any subroutines.

a. Lines 18 through 22

This section of code decrements the following by one:

<table>
<thead>
<tr>
<th>Column of PLATFM</th>
<th>Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>maximum speed</td>
</tr>
<tr>
<td>13</td>
<td>ADV</td>
</tr>
<tr>
<td>14</td>
<td>PDV</td>
</tr>
<tr>
<td>15</td>
<td>weapon one resupply</td>
</tr>
<tr>
<td>16</td>
<td>weapon two resupply</td>
</tr>
<tr>
<td>17</td>
<td>weapon three resupply</td>
</tr>
</tbody>
</table>

106
Column 11 of PLATFM contains durability point losses suffered. For ease in handling internally, it is carried as a negative value.

b. Lines 23 through 35

This section of code checks to see if the platform is still 'alive' and ensures it had not been sunk previously during the current game turn. If so, the program returns control to subroutine COMBAT. If the platform was sunk during this iteration, it locates the next available row in the SUNK array.

c. Lines 37 and 38

This section of code enters the row index of the sunk platform into SUNK and sets the durability value of the platform equal to zero (PLATFM(x,20)). This is the general indicator for the rest of the program. That is, if column 20 of PLATFM is zero for any platform, the platform is ignored for all practical purposes.

8. Subroutine MOVE

Subroutine MOVE executes all platform movement except platforms that have been sunk. It is called by subroutine CONTRL and does not call any subroutines.

a. Lines 14 through 17

This section of code replaces the current location (PLATFM(x,3)) with future location (PLATFM(x,5)). The DO 10 loop skips platforms that have been sunk.
9. **Subroutine SEARCH**

Subroutine SEARCH executes all phases of active and passive search missions after combat operations and movement have been executed. Note: Surface units cannot search against other surface units and aircraft cannot detect surface ships. It is called by subroutine CONTRL and call subroutines ADVSUB, AEVSUB, PDVSUB, PEVSUB and DETEK.

   a. Lines 23 through 36

   This section of code executes surface and air active search missions against subsurface platforms. If there are no surface or air platforms involved in the game, the program transfers to 45 (line 62). The DO 20 loop executes for aircraft and surface platforms with an active mission that are not sunk. The program calls subroutine ADVSUB to calculate the active detection value (ADV) for the searching platform. Then the DO 10 loop processes all submarines, bypassing those sunk, and calls subroutine AEVSUB to calculate the active evasion value (AEV) and range (DIST) between the two platforms. If the ADV is greater than or equal to the AEV, there is a detection and subroutine DETEK is called to file the detection in the DETECT array. The row indices of searcher and target, active/passive designation and range are passed.

   b. Lines 43 through 55

   This section of code is almost identical to the section described above except that it is the passive
search mission that is being executed. The passive detection value (PDV) and passive evasion value (PEV) are substituted for their active counterparts.

c. Lines 61 through 76

This section of code executes submarine active search missions against both surface ship and submarine platforms. It is coded in the same fashion as above except that the limits of the DO loops and indices are changed to reflect the differing platforms.

d. Lines 83 through 97

This section of code executes submarine passive search missions against surface ship and submarine platforms. It is coded in the same fashion as above except that the limits on the DO loops and indices have changed and it is executing passive vice active searches.

10. Subroutine ADVSUB(I)

This subroutine computes the active detection value (ADV) for a platform executing an active search mission. It utilizes I as an index to access the proper row of PLATFM. It is called by subroutine SEARCH and does not call any subroutines.

a. Line 15

Sets ADV equal to the active detection value for that platform.

b. Line 16 through 19

This section of code bypasses adjustments for damage and speed if the searching platform is an aircraft.
The ADV is decremented for both damage suffered and current speed of the platform.

c. Lines 20 through 22

This section of code decrements ADV for both erratic water conditions (ERATIC equaling one) and the platform being within one hex of a disturbed water zone (PLATFM(x,6) set to three). The ADV must be nonnegative.

11. Subroutine AEVSUB(I,J,DIST)

This subroutine calculates the active evasion value (AEV) for a potential target platform. I is the row index of the searching platform and J is the row index of the potential target. DIST is the distance between platforms I and J. It is called by SEARCH and calls DISTAN.

a. Lines 17 through 21

This section of code determines the distance between platforms I and J, then calculates the AEV by dividing the distance by three and rounding up. Note that Y is a real and the AEV is one even if the distance is zero. The resulting AEV is then increased if the target platform is in a disturbed water zone (line 21).

12. Subroutine PDVSUB(I)

This subroutine computes the passive detection value (PDV) for a platform executing a passive search mission. It utilizes I as an index to access the proper platform in PLATFM. It is called by subroutine SEARCH and does not call any subroutines.
a. Line 15
Sets PDV equal to the PDV value in PLATFM.

b. Lines 16 through 20
This section of code makes PDV adjustments for damage and current speed. If the platform is an aircraft (line 16) this section is bypassed. Also, if the platform does not suffer degradation of capability due to speed (line 18), this section is bypassed.

c. Lines 21 through 23
This section of code makes PDV adjustments for erratic water conditions (ERATIC equaling one) and for being within one hex of a disturbed water zone (PLATFM(x,6) is set to three). The PDV must be nonnegative.

13. Subroutine PEVSUB(I,J,DIST)
This subroutine calculates the passive evasion value (PEV) for a potential target platform. I is the row index of the searching platform and J is the row index of the potential target platform. DIST is the distance between platforms I and J. It is called by subroutine SEARCH and calls DISTAN.

a. Lines 17 through 21
This section of code determines the distance (DIST) between platforms I and J then calculates PEV by dividing by ten and rounding up. Note that Y is a real and the PEV is set to one even if the distance is zero. This PEV is then added to the evasion value (PLATFM(x,18)) and
and adjusted for being within one hex of a disturbed water zone.

b. Lines 22 through 24

This section of code adjusts the PEV downward if the platform executed a combat mission and/or was assigned an active search mission. The PEV must be nonnegative. If the target platform has a speed of zero, did not execute a combat mission and was not assigned an active search mission the PEV is defined to be six.

14. Subroutine DETEK(II,J,AP,DIST)

Subroutine DETEK files all detections in the detection array DETECT as they occur. II is the row index of the searching platform, J is the row index of the detected platform, AP is one if the detection was by active means and a zero if by passive means, and DIST is the distance between the platforms II and J. It is called by subroutine SEARCH and does not call any subroutines.

a. Lines 14 through 16

This section of code locates the next unused row of the array DETECT.

b. Lines 17 through 19

This section of code tells the players when over 100 detections have occurred, that detection and all future detections will not be processed as detections.
c. Lines 20 through 22

This section of code places the row index of the search platform in column one, the row index of the detected platform in column two, the active/passive indicator in column three and the distance in column four of the DETECT array.

I. UTILITY SUBROUTINES

The subroutines described in this section perform functions that can be considered "administrative" and not related to any particular part of the program previously discussed.

1. Subroutine NONVRT(INPUT, OUTPUT)

This subroutine takes INPUT which is a character string of numbers (70A1), discards any leading blanks and returns the integer form of the string as OUTPUT. It is used by the program to ease the entry of numbers that would otherwise require restrictive format statements. All variables in the subroutine are integer mode and three integer arrays are used. NUMBER(10) is a data array that contains the ten primary digits, zero through nine. X(70) is an array used for the temporary storage of the numbers in INPUT. BLANK and DOT are alpha variables that relate to a blank space and a decimal point. DO LOOP 1 is a loop that checks each column of INPUT for blank spaces. The first column number that contains a character is stored in IFNUM. DO LOOP 3 checks the columns IFNUM through 70 of INPUT for
the blank or decimal point that ends the character string. ILLNUM is set equal to this column number minus one. IFNUM and ILLNUM are the inclusive column numbers of INPUT that contain information, and INUM is the total number of columns that contain information. DO LOOP 6 runs from one to INUM with I as its index. DO LOOP 5 is nested in DO LOOP 6, and it compares each column of INPUT to the character digits in NUMBER. The number of the character that matches is placed in X(I). DO LOOP 7 runs from one to INUM with J as its index, and it actually computes the number placed in OUTPUT. INDEX is the number of the column of X being worked on, and it is decremented from right to left. OUTPUT is calculated by multiplying X(INDEX) by a factor of ten raised to the J minus one power. For example, the far right column would be multiplied by ten to the zero power, and added to OUTPUT. The third column from the right would be multiplied by ten to the second power or 100, and added to OUTPUT.

2. Subroutine INVRT1(INPUT,IPUT) and INVRT6(INPUT,IPUT)

These subroutines perform the same basic function which is to read a character string and return it without leading blanks. INVRT1 returns only the first character, and is used to record Y(es) and N(o) answers as "Y" and "N". It does this with one loop that assigns the first column of INPUT not equal to a blank space to IPUT. INVRT6 discards any leading blanks and returns the first six characters of a string. It is used for recording platform names. The
subroutine uses a nested loop with the outside loop searching for the first column of INPUT to contain information. The inside loop moves the first six columns of INPUT containing information to IPUT, and transfers control to the RETURN statement.

3. Subroutines SCLASS(NAME, INDEX, NAMEWP) and NAMEIN
   (INPUT, INDEX, NAME, AC1, AC2, SUR1, SUR2, SUB1, SUB2)

   These subroutines compare alpha names to the names in arrays, and return the row index numbers of the names that match. SCLASS searches the 130 class names in NAMESH with a simple search that just compares the first four columns of the arrays. If none of the names match, INDEX is returned as zero. NAMEIN searches the user supplied names located in the array NAME. This search is more complicated because the row index numbers are checked to see if they match the rows in PLATFM controlled by the calling player. This prevents the wrong index number being returned if both Blue and Red players use the same name for a platform.

4. Subroutine DISTAN(XX, YY, D)

   Subroutine DISTAN computes and returns a hex distance (D) between any two hex locations that are passed to it (XX, YY). It is called by subroutine STATAC, STATSS, CONTRL, FIRE, AEVSUB and PEVSUB. It does not call any subroutines.

   a. Lines 14 through 18

      This section of code separates the two hex locations XX and YY into their two components, the column
component the first two digits and the row component composed of the last two digits. It should be noted that this procedure is what limits the size of the game board. If the 100's in lines 14 through 17 were changed to 1000's, then the upper limit on the game board and in this program would nominally be 999 by 999. Actually, however, only the row index is limited to two digits. The column index is unlimited except for the hardware limit of the computer being utilized. D is initialized to zero and is the distance.

b. Lines 19 through 38

This section of code first compares the columns of the two locations. If they are equal then the distance is the absolute value of the difference in the row coordinates. If the column coordinates do not match, then through a series of comparisons, the row and column of the XX location is incremented or decremented to make it appear that the XX location is taking a 'step' closer to the YY location.

c. Lines 39 through 41

After the 'step' is taken above, the distance is incremented by one and the locations are compared column to column and row to row. If there is no difference in the coordinates in either location the subroutine returns the current value of D. Otherwise it transfers control to 100 (line 19) and section b above is repeated.
V. METHODOLOGY

A. INTRODUCTION

This chapter outlines a general structure that eases and provides a framework for converting tactical board games to computer-assisted war games. The structure presented is based on lessons learned during the conversion of "UP SCOPE!." It should be remembered that the program presented in Chapter IV is not the result of the outline described but just the opposite. Although the ideas presented here were largely incorporated in the program, the outline itself was not available during programming. The experience gained during this effort would result in some changes to the program if the programming effort was started again.

An effort to convert a board game to a computerized game must take into consideration the following five major areas:

1. The amount of computer assistance.
2. The rules of the game.
3. The procedures required by the game.
4. The structure of the data.
5. The input and output of information.

Although each of the above areas is related; they should, in general, be considered separately in the specific order presented.
B. COMPUTER ASSISTANCE

Computer assistance can be defined as the degree to which the computer performs the tasks previously required of the players. The assistance can range from just the calculation of hit probabilities all the way to graphic displays and voice input. The amount of computer assistance supplied by the game is dependent on the following three factors:

1. The Programmer

   The first factor involves the programmer. His ability and ideas concerning a computerized game will have a large effect on how much of the game he decides to program. However, this decision is usually made by the programmer's superiors.

2. The Budget

   One resource restriction involves the amount of money available for the effect. In this sense "money" includes equipment, as having money implies the ability to spend it on equipment. Budget considerations can largely be ignored by the programmer because he is given the parameters within which the assigned tasks must be accomplished. However, the manager must be aware of the tradeoffs between money and the computing capacity and speed of the computer, and how they will effect the amount of assistance the program can provide.
3. Time

The time factor involved is another resource restriction that can effect the amount of assistance provided by the program. The time factor can be driven by either the programmer or the player. Usually the degree of assistance is designed to minimize the player's time and not the programmer's, because a successful game will be played many times. An awkward game, which may have minimized the programmer's time, will waste more of the organization's personnel hours by forcing the players to play longer than need be.

4. Result

The assistance provided by the computerized version of Up Scope in relation to the board game is almost total. The only task required of the players, other than to be aware of the rules and information contained in the data summaries, is to maintain the game board with ship locations.

C. RULES AND PROCEDURES

The rules of a board game and its procedures are closely related. The difference lies in the idea that rules imply "can" or "cannot" and therefore "should" be adhered to in order to play the game correctly. Rules also usually involve some choice on the part of the players. They can be broken and yet the game can go on. Procedures on the other hand are the skeleton of the game and involve no choice on
the part of the players. They give the game its framework
around which the players make their decisions. An example
of a rule in Up Scope is limiting the number of hexes a ship
can move. A procedure would be the calculation of a hit
probability and the rolling of the die to determine the hit.
In a conversion effort the rules and procedures become
computer code and instructions. The programmer must become
extremely knowledgeable about them because his task is to
translate them to computer code with no accidental changes.

Changes in the rules and procedures can be the result
of two things. The first is the fact that the programmer is
restricted in some way that necessitates a change. The
second is to improve the game in some fashion. This
normally means a change to make the game more realistic.
Prior to coding the first computer statement, detailed flow
charts should be prepared for each aspect of the game.
These flowcharts will force the programmer to study and
learn the rules and procedures of the game. They will also
ensure the program is a "faithful" reproduction of the board
game, or highlight any changes that have been made.

Caution when dealing with rules and procedures is
required. Although they are closely related, rules involve
interpretation and choice. It should be remembered that the
conversion of a game to a computer program in effect makes
rules like procedures in that the player is confined to
acting in a certain manner. The game becomes more rigid and
self-umpiring. The purpose of programming a board game is to free the players from bookkeeping and computational chores but in the process it adds restrictions. The programmer should be aware of this tradeoff, and guard against an overly restrictive game.

The following are some of the changes made to Up Scope:

1. Weapons

SPI's procedure for handling the resupply of weapons involves a resupply number. This number is used to limit the number of weapons each platform can carry and fire. Each weapon can be fired once without a change in procedure, but after the first shot a qualifying roll of the die must be made to determine if the weapon can be fired again. If the resupply number is higher than the die, the weapon can be fired. If not, the weapon cannot be used until the next game turn when the procedure will be repeated. Every time the weapon is fired, its resupply number is reduced by one. The computer version of "UP SCOPE!" does not make the qualifying die roll described above because the procedure becomes too restrictive and unrealistic when the resupply number becomes small.

2. Hits

Another change designed to increase realism is the amount of information supplied to the players about weapon results. The SPI procedure gives the offensive player perfect information about each shot result. The
computerized version gives the player the probability of a hit or a detonation result if the weapon is a torpedo.

3. Aircraft

The main change concerning aircraft is the max speed of some of the aircraft. They were changed to reflect current capabilities.

D. DATA STRUCTURE

Once the amount of computer assistance has been determined and the flowcharts have been set, the data structure for the game can be finalized. Of course some consideration must be given to the data structure while the above is being accomplished, but it is critical that the data structure be finalized in its chronological place. The order should have the flowcharts prior to entering the first line of code. Too often coding is started prematurely in an attempt to get "something on paper". The proper sequence forces the programmer to consider all the game's data requirements before he commits himself to actual code. Any other procedure will result in numerous changes and additions.

Usually the data required by the game takes one of two forms. The first is basic data which includes game parameters, system specifications and assumptions. The basic data for Up Scope includes the data summaries in Appendices B, C and D. Information presented by a scenario can also be considered basic data. In general basic data
includes all the information that is available to the players. The second form of data consists of the working files used by the program to accomplish its tasks. The working files will be transparent to the players.

1. Basic Data

Basic data is often not a primary concern of the programmer because unlike the working files it is not manipulated or used extensively by the program. This should not be the case because it is possible to improve the program by careful structuring of the basic data. In the computer version of Up Scope, the basic data is entered using data statements. As previously mentioned, the ship data is entered using 18 different data arrays, one array for each separate piece of information. All of these arrays are equivalenced to SHIPDA which is used in the platform loading. Although this seems like a great deal of effort when one or two large data arrays could be used, it saved a great deal of time finding and correcting errors in coding. This was accomplished with no increase in computing time or storage requirements. Another aspect of the basic data structure which improved the program was the ordering of the countries' ships in SHIPDA. The order in SPI's rule book has the USSR first and the United States fifth. By itself this seems unimportant, but several times per game turn a search of the arrays to find a specific ship class is required. Under the assumption that players will use the
United States' and USSR's ships more often, the search times were reduced by placing the ships in that order.

2. Working Files

The working files are of major concern to the programmer because they provide a focus for his efforts. This is true in the sense that all the actions of the program are directed towards accepting, manipulating, retrieving and providing information. The rules and procedures are closely related to the structure of the working files, and careful thought about one can create a savings in the other. A good example of this is the format of the PLATFM array as previously shown in Figure 4. There were several ways to order the platforms. Blue and Red could have been in two groups, or the order of aircraft, surface ships and submarines could have been different. The present structure proved to be more efficient when considering the detection procedures. UP SCOPE! allows aircraft and surface ships to detect submarines, and it allows submarines to detect surface ships and other submarines. Surface ships cannot detect each other, and no platform can detect aircraft. The structure of PLATFM allows one loop to process each of the two detection phases. Other PLATFM formats would require numerous smaller loops to accomplish the same task. The structure of PLATFM also provides a counter example. The loading of the platforms is a relatively efficient section because it loads aircraft
automatically according to the basic data in Appendix B, but the fact that aircraft appear first in PLATFM required a great deal of index and pointer manipulation in the loading routine. It also resulted in a gap between aircraft and surface ships that had to be closed before the game could be started. All of these problems could have been avoided if the order of the platforms had been reversed. Having submarines first and aircraft last would have caused no change in the procedures, but would have greatly simplified the loading. Unfortunately once this was discovered too many changes in other parts of the program were required, and the effort would have been counter-productive.

The columns of PLATFM provide another double-edged example. The order of the information in each row is evidence that several additions were made after the fact. This has no effect at all on the program, and can be ignored from the player's point of view. One aspect of the row ordering that should not be ignored is the contiguous placement of certain types of information. For example, the placement of the ship's shot results in columns 35, 36 and 37 allow the use of one DO LOOP to report the results of all the weapons. Similar ordering is used for the weapon and aircraft index numbers and number of each type carried by the platform.
3. Limitations

Another aspect of data structures that should be considered by the programmer is the possible restrictions it imposes. All of the working file arrays in Up Scope are dimensioned to 100. This restricts the player from using more than 100 platforms in a game. The DETECT array imposes an even tighter restriction with its size of 100 because each platform can detect more than one ship. The choice of 100 was largely arbitrary, and can be changed with very little effort. It should also be noted that exceeding the limits of any array causes a message to be printed explaining the situation.

E. INPUT AND OUTPUT REQUIREMENTS

The amount and format of the information required and output by the program is decided largely by the degree of user-friendliness desired. The basic intent of this effort was to program a game that was very friendly. This resulted in numerous lines of output designed to give the player the ability to change his inputs and to clearly define his options when a selection was required from him. Unfortunately the overall effect is a program that is possibly too user-friendly because the amount of lines printed that contain information not directly related to the game is excessive. It actually slows the rate of play when a player becomes familiar with the game and its input requirements.
The above highlights the major tradeoff associated with input and output. A large amount of instructions and game information may be appropriate if the game will be played only a few times by each of a great number of players. However, a game that will be played many times by the same players should minimize the amount of extraneous information presented and requested.
VI. FUTURE ENHANCEMENTS AND CONCLUSION

A. FUTURE ENHANCEMENTS

1. Computer Assistance

   The computer assistance provided by the computerized game could be improved by the use of two terminals. This would allow each player his own console, and eliminate the need for passwords. It would also add an air of realism by physically separating the players. The present structure of the program is such that only minor changes would be required if the proper system support were available. Another improvement would be the use of graphic displays to present and maintain the game board for each player. This would eliminate the last "chore" required of the players.

2. Rules and Procedures

   Numerous recommendations could be made under this category. The only limitations are the size of the computer and the imagination of the programmer. Of the three presented here, the first two would have the effect of making the game more realistic. The third is strictly procedural.

   a. Land Bases

   The current version of Up Scope does not include the use of land based aircraft although the SPI version does and the aircraft are loaded in the data base. The
addition of land bases would require a special type of "platform" to be loaded in the PLATFM array. The loading of aircraft for this platform would have to be modified by allowing the player to choose the type of aircraft, unless the programmer designed several types of bases, each with different aircraft. These bases could be added to the database as "ship classes" without the ability to move or be sunk.

b. Surface Warfare

As designed by SPI, "UP SCOPE!" is a submarine warfare game. Future efforts might include the expansion of the game to include surface warfare as well. This type of change would involve a great deal of work because an entirely new game would result.

c. Game Board

One procedural improvement to the game would be computerized control of the playing areas. A check could be programmed to ensure neither player moved his ships beyond the area dictated by the scenario. This would eliminate the need for the "gentleman's agreement" to stay within the specified area.

3. Input and Output

Although the amount and format of input and output is determined by the type of player, the suggestions below would improve the game for both the expert player familiar with the instructions and the new player.
a. Platform Loading

The procedure for the initial loading of ships requires three separate lines of information to be input. All three pieces could be entered on one line with the use of a subroutine like NONVRT. The subroutine would be required to break a string of characters into three separate "words" with each word a different format or mode. These "words" could then be assigned to the appropriate variables. This procedure could be used in other parts of the program as well.

b. Aircraft loading

Every time a ship class that carries aircraft is loaded, the program automatically loads the aircraft and asks the player to input a name for each aircraft. This procedure becomes quite lengthy when more than a few aircraft are carried. It would decrease the loading time if the players were given the option of letting the program supply names for the aircraft based on the user-supplied name of the ship. For example, a Virginia class ship named GINNY could have its two Sea Sprites named GIN1 and GIN2. This change would not be too difficult to make, using character manipulation in the aircraft loading routine.

B. CONCLUSION

Although the outline presented in Chapter V does not provide a detailed checklist including step by step instructions for converting a tactical board game, it is
entirely sufficient. More detail would not necessarily be an improvement. As it is, the outline will provide the programmer contemplating a conversion a great deal of assistance. It supplies a structure that will guide and help organize his thoughts and efforts. If the outline had been available prior to the conversion of "UP SCOPE!", it would have resulted in the improvements discussed in Chapter V, and it would have saved a great deal of time spent reprogramming routines that had to be changed because they did not "fit" the rest of the program. The outline also provides a framework for future enhancements to the program.

The military student of war gaming and the programmer required to convert a board game will find this thesis highly beneficial. It not only provides a structure that will help the programmer, it gives the student some knowledge about designing, developing and programming war games that are highlighted by actually playing "UP SCOPE!."
APPENDIX A
SPECIFIC SCENARIOS

A. SCENARIO RESTRICTIONS

Each of these three scenarios constitutes a distinct game utilizing unique forces and special rules as required. A scenario normally ends at the conclusion of the full number of Game-Turns indicated in the scenario rules; however any of these three scenarios may be prematurely concluded during the victory determination phase (at the end of the game-turn) if any of the following conditions are satisfied.

1. All of the participating surface ships have been sunk or have exited the map boundaries.

2. All the submarines have either been sunk or have exited the map boundaries.

3. Either player has achieved a major victory and the opposition desires to terminate the game.

During the three scenarios that follow, one player is designated the "Attacker" and the other the "Defender". The attacker's mission is generally to inflict the maximum possible damage on the defender's platform(s). The defender's mission is generally to ensure that his platform(s) exit from the map boundary without suffering any damage and/or to inflict the maximum possible damage on the attacking platform(s). Either players platform's are permitted to exit the map boundaries but they may not later reenter the
game. It should be noted, however, that the defender receives final Victory Points for the durability value of each defending platform remaining on the map at the conclusion of the game, if there are no attacking platforms present on the map at the end of the game.

B. SCENARIO VICTORY DETERMINATION PROCEDURE

1. At the beginning of the game utilizing one of these three scenarios, the defender records the total durability value of all platform(s) under his control and records this Initial Victory Point Total.

2. At the end of the designated number of game turns, the defender records the total durability value for all platform(s) under his control which has exited the destination map boundary as specified in the scenario utilized.

3. The total derived above is incremented by one for each durability point of damage inflicted on the opposing platform(s) during the scenario played.

4. The total derived above is the final victory point total and is then divided by the initial victory point total and expressed as a percentage. This is the victory percentage ratio. The winner and level of victory achieved are then obtained from the following:
C. SCENARIOS

1. Here It Comes, Chicago!

a. Scenario description. On April 2, 1982, a Russian Delta class ballistic missile submarine is cruising off the northeast coast of the United States. For the last ten days the Soviet crew has been aware of, and hounded by, an American task force of two Spruance class destroyers and one Knox class escort. For the past 24 hours all platforms have been on special alert as the two superpowers approach the brink of nuclear exchange. Should hostilities commence, the Delta will attempt to obliterate a large chunk of real estate centered at Chicago!

b. Date: April 2, 1982
c. Water condition: Normal
d. Map format: 33 by 45 hexes
e. Game length: Ten Game-Turns
f. Special Rules:
   (1) During the command phase of each game turn, the U.S. player rolls a die keeping the result from the Soviet...
player. No platform may execute a combat mission until a six has been rolled. Once the six is rolled, all surface platforms may be assigned a combat mission according to the standard game rules. Once the U.S. player executes a combat mission, the Delta may be assigned combat missions according to the standard game rules.

(2) The Sea Sprite (Lamps) helicopters are not considered operational during this scenario.

(3) No platform may exit the board boundary during the scenario.

(4) If the Delta has not been sunk by the end of the tenth game turn, the U.S. player automatically loses both the game and Chicago.

g. Soviet/defender: One Delta class ballistic missile submarine; initial deployment anywhere within three hexes of hex 1723.

h. U.S./attacker: Two Spruances class destroyers, one Knox class escort deployed anywhere within 15 hexes of the western board boundary.

2. NATO and the Denmark Strait

a. Scenario Description: in order to interdict NATO shipping during periods of hostilities, Soviet submarines must transit several straits all of which are patrolled by NATO warships. An undeclared "hot" war commenced today, 26 June 1982.
A flotilla of Soviet submarines stationed in the Greenland Sea has been directed to move south of the Arctic circle for passage into the Atlantic. Most have orders to transit between Iceland and the Faeroe Islands. However, some attempt to make a more treacherous transit between Iceland and Greenland - through the Denmark Strait. NATO is rapidly deploying an ASW hunter-killer task group tasking them to intercept and destroy all submarines transiting the Denmark Strait. The task group consists of U.S., West German and Great Britain destroyers. This NATO task group must attempt to prevent the Soviets from making a successful passage into the Atlantic where their cruise missiles would wreak havoc on NATO shipping.

b. Date: June 28, 1982
c. Water condition: Normal
d. Map format: 33 by 90 hexes
e. Game length: 20 game turns
f. Special rules: The scenario automatically ends if at the end of any game turn either all the Soviet submarines have been sunk or exited off the western board boundary.

g. Soviet/defender: At the start of the scenario the Soviet player privately rolls a die three times, for each roll of four or less the player controls an Echo class submarine. The submarine mission is to transit the playing surface from east to west. The submarines start anywhere within five hexes of the eastern board boundary.
h. NATO/Attacker: One U.S. Spruance class destroyer, one West German Hamburg class destroyer and one Great Britain Leander class destroyer. The three are deployed anywhere within three hexes of the western board boundary.

3. **Sink the America**
   
a. Scenario description: Soviet submarines stalk U.S. aircraft carriers all over the world. In the event of hostilities, the Soviets can be expected to immediately engage our forward deployed big-deck aircraft carriers. In this scenario, the USS America is proceeding towards the Middle East war zone escorted by two modern cruisers and one escort.
   
b. Date: July 4, 1981
   
c. Water conditions: Erratic
   
d. Map format: 33 by 90 hexes
   
e. Game length: 20 game turns
   
f. Special rules: The aircraft aboard the USS America are not considered operational during this scenario. The helicopters aboard the cruisers and escort are operational. The scenario automatically ends during the victory determination phase if the USS America exits the eastern board boundary or if both Soviet submarines are sunk.
   
g. Soviet/attacker: Two Charlie class Soviet submarines deployed anywhere within ten hexes of the eastern board boundary.
h. U.S./defender: USS America, America class aircraft carrier deployed anywhere on the western board boundary, at least ten hexes from both northern and southern board boundaries. USS Virginia, Virginia class cruiser; USS South Carolina, Virginia class cruiser; USS T C Hart, Knox class escort; cruisers and escort are deployed within five hexes of the USS America.
APPENDIX B

SHIP DATA SUMMARY

<table>
<thead>
<tr>
<th>NO</th>
<th>SHIP CLASS</th>
<th>SP</th>
<th>AD</th>
<th>PD</th>
<th>SS</th>
<th>SD</th>
<th>EV</th>
<th>DE</th>
<th>DU</th>
<th>NO-WEAPON</th>
<th>NO-AIRCRAFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NIMITZ</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>6-SEA KING</td>
<td>6-VIKING</td>
</tr>
<tr>
<td>2</td>
<td>AMERICA</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>6-SEA KING</td>
<td>6-VIKING</td>
</tr>
<tr>
<td>3</td>
<td>KITTY HAWK</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>6-SEA KING</td>
<td>6-VIKING</td>
</tr>
<tr>
<td>4</td>
<td>FORESTAL</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>6-SEA KING</td>
<td>6-VIKING</td>
</tr>
<tr>
<td>5</td>
<td>MIDWAY</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>6-SEA KING</td>
<td>6-VIKING</td>
</tr>
<tr>
<td>6</td>
<td>ENTERPRISE</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>6-SEA KING</td>
<td>6-VIKING</td>
</tr>
<tr>
<td>7</td>
<td>VIRGINIA</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4-ASROC</td>
<td>4-MK 46</td>
</tr>
<tr>
<td>8</td>
<td>CALIFORNIA</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>5-ASROC</td>
<td>3-MK 46</td>
</tr>
</tbody>
</table>

UNITED STATES AIRCRAFT CARRIERS

CRUISERS

7 VIRGINIA 3 5 5 1 1 4 3 2 4-ASROC 4-MK 46

8 CALIFORNIA 3 5 4 1 1 4 3 2 5-ASROC 3-MK 46

2-SEA SPRITE

139
<table>
<thead>
<tr>
<th>NO</th>
<th>SHIP CLASS</th>
<th>SP</th>
<th>AQ</th>
<th>PO</th>
<th>SS</th>
<th>SD</th>
<th>EV</th>
<th>DE</th>
<th>DU</th>
<th>NO-WEAPON</th>
<th>NO-AIRCRAFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>TRUXTON</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>5=ASROC</td>
<td>2=SEA SPRITE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3=MK 46</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>BELKNAP</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3=ASROC</td>
<td>1=SEA SPRITE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>BAINBRIDGE</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>5=ASROC</td>
<td>1=SEA SPRITE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>LEAHY</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>5=ASROC</td>
<td>1=SEA SPRITE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ALBANY</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>5=ASROC</td>
<td>1=SEA SPRITE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>CHICAGO</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>5=ASROC</td>
<td>1=SEA SPRITE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>LONG BEACH</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>5=ASROC</td>
<td>1=SEA SPRITE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>CLEVELAND</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DESTROYERS**

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>REGIS</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>5=ASROC</td>
<td>2=LAMPS MK III</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>SPRUANCE</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4=MK 46</td>
<td>1=SEA SPRITE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5=ASROC</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>NITSCHE</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5=ASROC</td>
<td>4=MK 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>FORREST SHERMAN</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5=ASROC</td>
<td>4=MK 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>COONTZ</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>5=ASROC</td>
<td>4=MK 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>CHARLES F. ADAMS</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5=ASROC</td>
<td>4=MK 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>CARPENTER</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5=ASROC</td>
<td>4=MK 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>GEARING</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5=ASROC</td>
<td>4=MK 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>SHIP CLASS</td>
<td>SP</td>
<td>RD</td>
<td>PD</td>
<td>SS</td>
<td>SD</td>
<td>EV</td>
<td>DE</td>
<td>DU</td>
<td>NO=weapon</td>
<td>NO=aircraft</td>
</tr>
<tr>
<td>----</td>
<td>------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>25</td>
<td>OLIVER II. PEARL</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=mk 48</td>
<td>2=sea sprite</td>
</tr>
<tr>
<td>26</td>
<td>KNOX</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5=asroc</td>
<td>1=sea sprite</td>
</tr>
<tr>
<td>27</td>
<td>BROOKE</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5=asroc</td>
<td>1=sea sprite</td>
</tr>
<tr>
<td>28</td>
<td>GARCIA</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5=asroc</td>
<td>1=sea sprite</td>
</tr>
<tr>
<td>29</td>
<td>TRIDENT</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>4=mk 48</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>LAFAYETTE</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4=mk 48</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>ETHAN ALLEN</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4=mk 48</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>G. WASHINGTON</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4=mk 48</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>LOS ANGELES</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>3=subroc</td>
<td>4=mk 48</td>
</tr>
<tr>
<td>34</td>
<td>NARWHAL</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>3=subroc</td>
<td>4=mk 48</td>
</tr>
<tr>
<td>35</td>
<td>STURGEON</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>3=subroc</td>
<td>4=mk 48</td>
</tr>
<tr>
<td>36</td>
<td>PERMIT</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3=subroc</td>
<td>4=mk 48</td>
</tr>
<tr>
<td>37</td>
<td>TULIBEE</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>3=subroc</td>
<td>4=mk 48</td>
</tr>
<tr>
<td>38</td>
<td>SKIPJACK</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3=subroc</td>
<td>4=mk 48</td>
</tr>
<tr>
<td>39</td>
<td>SKATE</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3=subroc</td>
<td>4=mk 48</td>
</tr>
<tr>
<td>NO</td>
<td>SHIP CLASS</td>
<td>SP AD PD SS SD EV DE DU</td>
<td>NO. WEAPON</td>
<td>NO. AIRCRAFT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>KIEV</td>
<td>3 4 5 1 1 2 3 3 5=SSN-14</td>
<td>10=HORMONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4=21&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>MOSKVA</td>
<td>3 4 5 1 1 3 2 2=FAAS-1</td>
<td>5=HORMONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4=21&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>KARA</td>
<td>3 4 5 1 1 3 2 2=SSN-14</td>
<td>1=HORMONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>KRESTA</td>
<td>3 3 4 0 0 3 2 2=SSN-14</td>
<td>1=HORMONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>KYNDA</td>
<td>4 3 4 0 0 3 2 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>KRAIAK</td>
<td>4 4 5 1 1 4 2 1=SSN-14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>KANIN</td>
<td>4 3 4 0 0 4 2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>KOTLIN</td>
<td>4 3 4 0 0 4 2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>KASHIN</td>
<td>4 3 5 1 1 4 2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6=MBU-4500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>SKORY</td>
<td>3 3 4 0 0 4 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>MIRKA</td>
<td>3 3 4 0 0 4 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4=16&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>PETTA</td>
<td>3 3 4 0 0 4 2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7=MBU-2500A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO SHIP CLASS</td>
<td>SP</td>
<td>AD</td>
<td>PD</td>
<td>SS</td>
<td>SD</td>
<td>EV</td>
<td>DE</td>
<td>GU</td>
<td>NO=WEAPON</td>
<td>NO=AIRCRAFT</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>52 RIGA</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>6=Mbu-2500A</td>
<td>4=OC</td>
<td></td>
</tr>
</tbody>
</table>

**SUBMARINES (BALLISTIC MISSILE):**

<table>
<thead>
<tr>
<th>NO</th>
<th>Class</th>
<th>SP</th>
<th>AD</th>
<th>PD</th>
<th>SS</th>
<th>SD</th>
<th>EV</th>
<th>DE</th>
<th>GU</th>
<th>NO=WEAPON</th>
<th>NO=AIRCRAFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>Delta</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4=21&quot;</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>Yankee</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4=21&quot;</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>Golf</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>4=21&quot;</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>Hotel</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4=21&quot;</td>
<td></td>
</tr>
</tbody>
</table>

**SUBMARINES (ATTACK):**

<table>
<thead>
<tr>
<th>NO</th>
<th>Class</th>
<th>SP</th>
<th>AD</th>
<th>PD</th>
<th>SS</th>
<th>SD</th>
<th>EV</th>
<th>DE</th>
<th>GU</th>
<th>NO=WEAPON</th>
<th>NO=AIRCRAFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>Charlie</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1=SSN-7</td>
<td>4=21&quot;</td>
</tr>
<tr>
<td>58</td>
<td>Victor</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4=21&quot;</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>Echo</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1=SSN-3</td>
<td>4=21&quot;</td>
</tr>
<tr>
<td>60</td>
<td>November</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3=21&quot;</td>
<td></td>
</tr>
</tbody>
</table>

**GREAT BRITAIN CRUISERS:**

<table>
<thead>
<tr>
<th>NO</th>
<th>Class</th>
<th>SP</th>
<th>AD</th>
<th>PD</th>
<th>SS</th>
<th>SD</th>
<th>EV</th>
<th>DE</th>
<th>GU</th>
<th>NO=WEAPON</th>
<th>NO=AIRCRAFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>61</td>
<td>Invincible</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>2</td>
<td>2</td>
<td>10=sea king</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>Bristol</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3=Ikarra</td>
<td>6=Limbo</td>
</tr>
</tbody>
</table>

**DESTROYERS:**

<table>
<thead>
<tr>
<th>NO</th>
<th>Class</th>
<th>SP</th>
<th>AD</th>
<th>PD</th>
<th>SS</th>
<th>SD</th>
<th>EV</th>
<th>DE</th>
<th>GU</th>
<th>NO=WEAPON</th>
<th>NO=AIRCRAFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>Weapon</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=Mk 46</td>
<td>2=Ltnx</td>
</tr>
<tr>
<td>64</td>
<td>Sheffield</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=Mk 46</td>
<td>1=Ltnx</td>
</tr>
<tr>
<td>65</td>
<td>Amazon</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=Mk 46</td>
<td>1=Ltnx</td>
</tr>
<tr>
<td>66</td>
<td>Leander</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>6=Limbo</td>
<td>4=MK 46</td>
</tr>
<tr>
<td>67</td>
<td>Rothesay</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>6=Limbo</td>
<td>1=Wasp</td>
</tr>
<tr>
<td>NO</td>
<td>SHIP CLASS</td>
<td>SP</td>
<td>AD</td>
<td>PD</td>
<td>SS</td>
<td>SO</td>
<td>EV</td>
<td>DE</td>
<td>DU</td>
<td>NO=WEAPON</td>
<td>NO=AIRCRAFT</td>
</tr>
<tr>
<td>----</td>
<td>------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>68</td>
<td>TRIBAL</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>6=LIMBO</td>
<td>1=WASP</td>
</tr>
<tr>
<td>69</td>
<td>LEOPARD</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>6=LIMBO</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>SALISBURY</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>6=LIMBO</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>BLACKWOOD</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>6=LIMBO</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>WHITBY</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>6=LIMBO</td>
<td></td>
</tr>
</tbody>
</table>

**SUBMARINES (BALLISTIC MISSILE)**

<table>
<thead>
<tr>
<th>NO</th>
<th>RESOLUTION</th>
<th>2</th>
<th>4</th>
<th>4</th>
<th>0</th>
<th>0</th>
<th>4</th>
<th>2</th>
<th>2</th>
<th>4=TIGERFISH</th>
</tr>
</thead>
</table>

**SUBMARINES (ATTACK)**

<table>
<thead>
<tr>
<th>NO</th>
<th>SWIFTSURE</th>
<th>3</th>
<th>4</th>
<th>0</th>
<th>0</th>
<th>4</th>
<th>2</th>
<th>2</th>
<th>4=TIGERFISH</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>VALIANT</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>76</td>
<td>DREADNOUGHT</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>77</td>
<td>OBERON</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**CANADA**

**DESTROYERS**

<table>
<thead>
<tr>
<th>NO</th>
<th>TRIBAL</th>
<th>3</th>
<th>4</th>
<th>4</th>
<th>1</th>
<th>1</th>
<th>4</th>
<th>2</th>
<th>1</th>
<th>4=MK 46</th>
</tr>
</thead>
<tbody>
<tr>
<td>79</td>
<td>ANNAPOLIS</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=MK 46</td>
</tr>
<tr>
<td>80</td>
<td>ST. LAURENT</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=MK 46</td>
</tr>
<tr>
<td>81</td>
<td>HACKERIE</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=MK 46</td>
</tr>
<tr>
<td>82</td>
<td>RESTINGOUCHE</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=MK 46</td>
</tr>
<tr>
<td>NO</td>
<td>SHIP CLASS</td>
<td>SP</td>
<td>R</td>
<td>PO</td>
<td>SS</td>
<td>SD</td>
<td>EV</td>
<td>DE</td>
<td>DU</td>
<td>NO=WEAPON</td>
</tr>
<tr>
<td>----</td>
<td>------------</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-------------</td>
</tr>
<tr>
<td>SUBMARINES (1 HITCOK)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83 OBERON</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4=TIGERFISH</td>
<td>4=MK 20</td>
</tr>
<tr>
<td>AUSTRALIA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIRCRAFT CARRIERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84 MELBOURNE</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td>5=SEA KING</td>
<td></td>
</tr>
<tr>
<td>DESTROYERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85 PERTH</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=IKARA</td>
<td></td>
</tr>
<tr>
<td>86 DARING</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4=LIMBO</td>
<td></td>
</tr>
<tr>
<td>FRIGATES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>87 O.H. PÉRIGUEUX</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4=MK 46</td>
<td></td>
</tr>
<tr>
<td>88 RIVER</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>3=IKARA</td>
<td>4=LIMBO</td>
</tr>
<tr>
<td>SUBMARINES (1 TACK)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89 OBERON</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4=TIGERFISH</td>
<td>4=MK 20</td>
</tr>
<tr>
<td>FRANCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIRCRAFT CARRIERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90 CLEMENCEAU</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>10=ALIZE</td>
<td>5=SUPER FRELON</td>
</tr>
<tr>
<td>CRUISERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91 JEANNE D'ARC</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>6=SUPER FRELON</td>
<td></td>
</tr>
<tr>
<td>92 COLBERT</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Ship Class</td>
<td>SP</td>
<td>AD</td>
<td>PD</td>
<td>SS</td>
<td>SD</td>
<td>DE</td>
<td>EV</td>
<td>DE</td>
<td>DU</td>
</tr>
<tr>
<td>----</td>
<td>------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>93</td>
<td>Montcalm</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>5MK</td>
</tr>
<tr>
<td>94</td>
<td>DuPere</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>5MK</td>
</tr>
<tr>
<td>95</td>
<td>Tourville</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>5MK</td>
</tr>
<tr>
<td>96</td>
<td>Aconit</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>5MK</td>
</tr>
<tr>
<td>97</td>
<td>Suffren</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>5MK</td>
</tr>
<tr>
<td>98</td>
<td>La Galissonniere</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4MK</td>
</tr>
<tr>
<td>99</td>
<td>Tartu</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4MK</td>
</tr>
<tr>
<td>100</td>
<td>Ker SAINT</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4MK</td>
</tr>
<tr>
<td>101</td>
<td>Casablanca</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4MK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6MK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6MK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6MK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6MK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6MK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6MK</td>
</tr>
<tr>
<td>102</td>
<td>Destienne</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4MK</td>
</tr>
<tr>
<td>103</td>
<td>Comm, Riviere</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4MK</td>
</tr>
<tr>
<td>104</td>
<td>Savoyard</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4MK</td>
</tr>
<tr>
<td>105</td>
<td>Le Boulonnais</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4MK</td>
</tr>
<tr>
<td>NO SHIP CLASS</td>
<td>SP AD PD</td>
<td>SS SD</td>
<td>EV DE</td>
<td>DU</td>
<td>NO-WEAPON</td>
<td>NO-AIRCRAFT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>------</td>
<td>------</td>
<td>---</td>
<td>-----------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SUBMARINES (BALLISTIC MISSILE)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>106 REDOUTABLE</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>S=MK E12</td>
<td></td>
</tr>
<tr>
<td><strong>SUBMARINES (ATTACK)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>107 AGOSTA</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>S=MK E12</td>
<td></td>
</tr>
<tr>
<td>108 DAPHNE</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>S=MK E12</td>
<td></td>
</tr>
<tr>
<td>109 ARETHUSE</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>S=MK E12</td>
<td></td>
</tr>
<tr>
<td>110 NARVAL</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>S=MK E12</td>
<td></td>
</tr>
<tr>
<td><strong>ITALY</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CRUISERS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111 VITTORIO VENETO</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>S=MK 44</td>
<td></td>
</tr>
<tr>
<td>112 ANDREA DORIA</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>S=MK 44</td>
<td></td>
</tr>
<tr>
<td><strong>DESTROYERS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>113 AUDACE</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>S=MK 44</td>
<td></td>
</tr>
<tr>
<td>114 IMPAVIDO</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>S=MK 44</td>
<td></td>
</tr>
<tr>
<td>115 IMPETUOSO</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>S=MK 44</td>
<td></td>
</tr>
<tr>
<td><strong>FRIGATES</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>116 LUPO</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>S=MK 44</td>
<td></td>
</tr>
<tr>
<td>117 ALPINO</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>S=MK 44</td>
<td></td>
</tr>
<tr>
<td>118 DE CHRISTOFARO</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>S=MK 44</td>
<td></td>
</tr>
<tr>
<td>119 BERGAMINI</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>S=MK 44</td>
<td></td>
</tr>
<tr>
<td>120 CENTAURUS</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>S=MK 44</td>
<td></td>
</tr>
</tbody>
</table>

147
<table>
<thead>
<tr>
<th>No</th>
<th>Ship Class</th>
<th>SP AO</th>
<th>PD</th>
<th>SS</th>
<th>SD</th>
<th>EV</th>
<th>DE</th>
<th>DU</th>
<th>No=Weapon</th>
<th>No=AircrAft</th>
</tr>
</thead>
<tbody>
<tr>
<td>121</td>
<td>ALBATROS</td>
<td>2 0 0 0 4 0 1</td>
<td>5=MK 44</td>
<td>6=HEDGEHOG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>SAURO</td>
<td>2 3 4 0 5 2 2</td>
<td>5=533MM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>EX-TANG</td>
<td>2 3 4 0 5 1 2</td>
<td>4=533MM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>TOTI</td>
<td>2 3 4 0 5 1 2</td>
<td>5=533MM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>LUTJENS</td>
<td>4 4 0 0 4 2 1</td>
<td>5=MK 44</td>
<td>5=ASROC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>HAMBURG</td>
<td>4 3 4 0 4 2 1</td>
<td>4=533MM</td>
<td>6=BOFORS 375MM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>KOLN</td>
<td>3 3 4 0 4 2 1</td>
<td>4=533MM</td>
<td>6=BOFORS 375MM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>THETIS</td>
<td>3 3 4 0 4 1 1</td>
<td>4=533MM</td>
<td>6=BOFORS 375MM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>U-206</td>
<td>2 0 4 0 5 1 2</td>
<td>5=MK 37C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>U-205</td>
<td>2 0 4 0 5 1 2</td>
<td>5=MK 37C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C

WEAPON DATA SUMMARY

<table>
<thead>
<tr>
<th>NO</th>
<th>NAME</th>
<th>ACCURACY</th>
<th>RANGE</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SSN-3 (USSR)</td>
<td>5</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>SSN-7 (USSR)</td>
<td>5</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>SUBROC (US)</td>
<td>6</td>
<td>25</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>SSN-14 (USSR)</td>
<td>6</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>21&quot; (USSR)</td>
<td>5</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>16&quot; (USSR)</td>
<td>5</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>MK 48 (US)</td>
<td>6</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>MK 37C (US, HQ)</td>
<td>5</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>TIGERFISH (GB)</td>
<td>6</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>MK 20 (GB)</td>
<td>5</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>MK K2 (FA)</td>
<td>6</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>MK L5 (FA)</td>
<td>5</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>MK E12 (FA)</td>
<td>5</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>533MM (IT)</td>
<td>5</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>ASROC (US, GB)</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>MK 46 (US)</td>
<td>5</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>MK 44 (US, GB, IT)</td>
<td>4</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>1KARA (GB, AUS)</td>
<td>6</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>19</td>
<td>MALAFON (FA)</td>
<td>5</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>NO</td>
<td>NAME</td>
<td>ACCURACY</td>
<td>RANGE</td>
<td>TYPE</td>
</tr>
<tr>
<td>----</td>
<td>-------------------</td>
<td>----------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>20</td>
<td>FRAS-1 (USSR)</td>
<td>6</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>MBU-2500A (USSR)</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>MBU-4500A (USSR)</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>LIMBO (GB)</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>24</td>
<td>MEGHOG (GB)</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>25</td>
<td>375MM (FRA)</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>305MM (FRA)</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>27</td>
<td>MENON (IT)</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>28</td>
<td>BOFORS 375MM (GB)</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>29</td>
<td>DC (ALL NATIONS)</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
## APPENDIX D

### AIRCRAFT DATA SUMMARY

<table>
<thead>
<tr>
<th>NO.</th>
<th>NAME</th>
<th>SP</th>
<th>AD</th>
<th>PD</th>
<th>NO.</th>
<th>NAME</th>
<th>SP</th>
<th>AD</th>
<th>PD</th>
<th>NO.</th>
<th>NAME</th>
<th>SP</th>
<th>AD</th>
<th>PD</th>
<th>NO.</th>
<th>NAME</th>
<th>SP</th>
<th>AD</th>
<th>PD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>INDEX NUMBER OF THE AIRCRAFT TYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NAME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAME OF THE AIRCRAFT TYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MAX SPEED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ACTIVE DETECTION VALUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PASSIVE DETECTION VALUE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO=WEPON: NUMBER AND TYPE OF WEAPON CARRIED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>HORSE (USSA)</td>
<td>11</td>
<td>4</td>
<td>4</td>
<td>1=MK 46</td>
<td>3=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SEA KING (NATO)</td>
<td>11</td>
<td>4</td>
<td>4</td>
<td>1=MK 46</td>
<td>1=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>LYNX (NATO)</td>
<td>15</td>
<td>4</td>
<td>4</td>
<td>1=MK 46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>WASP (NATO)</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>1=MK 44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>SEA SPIRE (US)</td>
<td>12</td>
<td>4</td>
<td>4</td>
<td>1=MK 46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>LAMPS MK III (US)</td>
<td>16</td>
<td>4</td>
<td>4</td>
<td>1=MK 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>WESSEX (GB)</td>
<td>12</td>
<td>3</td>
<td>3</td>
<td>1=MK 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>SUPER FRELON (FRA)</td>
<td>14</td>
<td>4</td>
<td>4</td>
<td>1=MK 44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>AB-204 (IT)</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>1=MK 44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>MAIL (USSA)</td>
<td>24</td>
<td>3</td>
<td>3</td>
<td>4=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>MAY (USSA)</td>
<td>31</td>
<td>3</td>
<td>3</td>
<td>5=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>TRACKER (US)</td>
<td>15</td>
<td>4</td>
<td>4</td>
<td>1=MK 46</td>
<td>3=DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**151**
<table>
<thead>
<tr>
<th>NO</th>
<th>NAME</th>
<th>SP</th>
<th>RD</th>
<th>PV</th>
<th>NO</th>
<th>WEAPON</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>ORION (US)</td>
<td>30</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>MK 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 = DC</td>
</tr>
<tr>
<td>14</td>
<td>VIKING (US)</td>
<td>35</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>MK 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 = DC</td>
</tr>
<tr>
<td>15</td>
<td>NIMROD (US)</td>
<td>35</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>MK 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 = DC</td>
</tr>
<tr>
<td>16</td>
<td>ALIZE (FR)</td>
<td>24</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>MK 44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 = DC</td>
</tr>
</tbody>
</table>
APPENDIX E

PRACTICE GAME

> $ scope
EXECUTION BEGINS...

THE FOLLOWING OPTIONS ARE AVAILABLE TO YOU:

1. ZERO ANY PREVIOUSLY ESTABLISHED DATA BASE.
2. EXIT THE GAME.
3. INITIALIZE GAME PARAMETERS AND COMMENCE PLAY.

ENTER AN OPTION.
> 3

YOU ARE NOW GOING TO ENTER THE BASIC GAME PARAMETERS.
ONE WORD OF CAUTION. THE PRESENT GAME CAN ONLY HANDLE A
TOTAL OF 100 WEAPON SYSTEMS. INCLUDED IN THESE SYSTEMS
ARE SURFACE SHIPS, SUBMARINES AND AIRCRAFT. THE NUMBER OF
AIRCRAFT ARE DETERMINED BY THE SHIP CLASS AS REFLECTED IN THE
DATA BASE, AND WILL BE LOADED AUTOMATICALLY WHEN YOU SELECT
THE SHIP CLASS. PRIOR TO BEGINNING THE LOADING PROCEDURE
YOU SHOULD ENSURE THAT THE TOTAL NUMBER OF EACH PLAYERS
WEAPON SYSTEMS DOES NOT EXCEED 50.

ENTER THE NUMBER OF GAME TURNS YOU WANT TO PLAY.
> 2

ENTER THE WATER CONDITION (0-NORMAL, 1-ERRATIC).
> 0

THE FOLLOWING INPUTS APPLY ONLY TO THE BLUE PLAYER
THE FIRST IS THE BLUE PLAYERS PASSWORD WHICH IS
A MAXIMUM OF 6 CHARACTERS. THE PASSWORD IS IMPORTANT BECAUSE
IT IS REQUIRED PRIOR TO THE INPUT OF COMBAT, MOVE AND SEARCH
COMMANDS. IF YOU FORGET THE PASSWORD, YOU WILL HAVE TO RETURN
TO MENU SELECTION, AND RESTART THE GAME.

SELECT AND ENTER YOUR PASSWORD
> blupas
YOUR PASSWORD WILL BE -BLUPAS-. IF YOU ARE READY TO CONTINUE TYPE Y(ES). TYPE N(O) IF YOU WISH TO CHANGE YOUR PASSWORD.

> y

ENTER THE NUMBER OF BLUE SURFACE SHIPS.

> 2

ENTER THE NUMBER OF BLUE SUBMARINES.

> 1

ENTER THE NUMBER OF BLUE SHIPS THAT MUST BE DESTROYED FOR PREMATURE GAME ENDING

> 2

ECHO

BLUE SURFACE SHIPS - 2
BLUE SUBMARINES - 1
NUMBER OF BLUE SHIPS DESTROYED TO END GAME - 2

TYPE Y(ES) IF THESE VALUES ARE CORRECT, OR N(O) IF YOU WANT TO REENTER THEM.

> y

THE FOLLOWING INPUTS APPLY ONLY TO THE RED PLAYER.
THE FIRST IS THE RED PLAYERS PASSWORD WHICH IS A MAXIMUM OF 6 CHARACTERS. THE PASSWORD IS IMPORTANT BECAUSE IT IS REQUIRED PRIOR TO THE INPUT OF COMBAT, MOVE AND SEARCH COMMANDS. IF YOU FORGET THE PASSWORD, YOU WILL HAVE TO RETURN TO MENU SELECTION, AND RESTART THE GAME.

SELECT AND ENTER YOUR PASSWORD.

> redpas

YOUR PASSWORD WILL BE -REDPAS-. IF YOU ARE READY TO CONTINUE TYPE Y(ES). TYPE N(O) IF YOU WISH TO CHANGE YOUR PASSWORD.

> y.

ENTER THE NAME OF RED SURFACE SHIPS.

> 0

154
ENTER THE NUMBER OF RED SUBMARINES.
> 2

ENTER THE NUMBER OF RED SHIPS THAT MUST BE DESTROYED FOR PREMATURE GAME ENDING.
> 2

ECHO

RED SURFACE SHIPS - 0
RED SUBMARINES - 2
NUMBER OF RED SHIPS DESTROYED TO END GAME - 2

TYPE Y(ES) IF THESE VALUES ARE CORRECT, OR N(O) IF YOU WANT TO REENTER THEM.
> y

YOU ARE NOW READY TO LOAD THE BLUE WEAPON SYSTEMS.

ENTER THE BLUE PASSWORD.
> blupas

BLUE SURFACE SHIP NUMBER 1

ENTER A 6 CHARACTER NAME FOR THE SHIP.
> tincan

ENTER THE SHIP CLASS OF TINCAN
> spruance

ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF TINCAN
> 500

THE SPRUANCE CLASS OF SHIP HAS 1 SEA SPRITE TYPE AIRCRAFT.

ENTER A 6 CHARACTER NAME FOR AIRCRAFT 1
> sprul

BLUE SURFACE SHIP NUMBER 2

ENTER A 6 CHARACTER NAME FOR THE SHIP.
> escort

ENTER THE SHIP CLASS OF ESCORT
> knox
ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF ESCORT
> 520

THE KNOX CLASS OF SHIP HAS 1 SEA SPRITE TYPE AIRCRAFT.

ENTER A 6 CHARACTER NAME FOR AIRCRAFT 1
> knoxl

BLUE SUBMARINE NUMBER 1

ENTER A 6 DIGIT CHARACTER NAME FOR THE SUBMARINE.
> lapon

ENTER THE SUBMARINE CLASS OF LAPON
> sturgeon

ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF LAPON
> 510

YOU ARE NOW READY TO LOAD THE RED WEAPON SYSTEMS.

ENTER THE RED PASSWORD.
> redpas

RED SUBMARINE NUMBER 1

ENTER A 6 DIGIT CHARACTER NAME FOR THE SUBMARINE.
> ivan

ENTER THE SUBMARINE CLASS OF IVAN
> charlie

ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF IVAN
> 1510

RED SUBMARINE NUMBER 2

ENTER A 6 DIGIT CHARACTER NAME FOR THE SUBMARINE.
> boris

ENTER THE SUBMARINE CLASS OF BORIS
> november

ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL LOCATION OF BORIS
> 1520

156
BLUE CALL TO STATUS ROUTINES. ENTER THE BLUE PASSWORD.
> blupas

THE FOLLOWING DISPLAY OPTIONS ARE AVAILABLE TO YOU:

1. DETECTIONS OF ENEMY PLATFORMS.
2. RESULTS OF WEAPONS FIRED.
3. DAMAGE TO FRIENDLY PLATFORMS.
4. STATUS OF FRIENDLY PLATFORMS AND INPUT OF INSTRUCTIONS.
5. BEGIN NEXT GAME TURN.

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.
> 4

YOU HAVE SELECTED INDIVIDUAL PLATFORM STATUS DISPLAY AND
ASSOCIATED COMBAT, MOVEMENT AND SEARCH COMMAND INPUTS.
ENTER THE NAME OF THE DESIRED PLATFORM.

>tincan

TINCAN IS LOCATED AT POSITION 500 SPEED 0
IT WAS NOT ASSIGNED A SEARCH MISSION
TINCAN WEAPON SYSTEM(S)
   TYPE       RANGE
   1 MK 46    20
   2 ASROC    7

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER \ IN THIS FASHION

>n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION \ IN THAT ORDER, OTHERWISE ENTER N(ONE)
> 2
> 502
ENTER SEARCH MISSIONS AS FOLLOWS:
   0  NONE SCHEDULED
   1  ACTIVE MISSION
   2  PASSIVE MISSION
   3  ACTIVE AND PASSIVE

> 3

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
   NONE SCHEDULED
DESTINATION 502  SPEED 2
ACTIVE AND PASSIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> y

ENTER THE NAME OF THE DESIRED PLATFORM

> escort

ESCORT IS LOCATED AT POSITION 520  SPEED 0
IT WAS NOT ASSIGNED A SEARCH MISSION
ESCORT WEAPON SYSTEM(S)
   TYPE   RANGE
   1  ASROC   7
   2  MK 46   20

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

> n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)
ENTER SEARCH MISSIONS AS FOLLOWS:
0  NONE SCHEDULED
1  ACTIVE MISSION
2  PASSIVE MISSION
3  ACTIVE AND PASSIVE

> 1

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
  NONE SCHEDULED
DESTINATION 720  SPEED 2
ACTIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION

> y

ENTER THE NAME OF THE DESIRED PLATFORM

> lapon

LAPON IS LOCATED AT POSITION 510  SPEED 0
IT WAS NOT ASSIGNED A SEARCH MISSION
LAPON WEAPON SYSTEM(S)
  TYPE   RANGE
  1  SUBROC   25
  2  MK 48     25

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER    IN THIS FASHION

> n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION    IN THAT ORDER, OTHERWISE ENTER N(ONE)

> 2
> 512
ENTER SEARCH MISSIONS AS FOLLOWS:
  0  NONE SCHEDULED
  1  ACTIVE MISSION
  2  PASSIVE MISSION
  3  ACTIVE AND PASSIVE

> 2

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
  NONE SCHEDULED
  DESTINATION 512 SPEED 2
  PASSIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION

> y

ENTER THE NAME OF THE DESIRED PLATFORM

> knox1

KNOX1 IS CURRENTLY NOT FLYING - IT IS ASSUMED YOU DESIRED TO LAUNCH.

KNOX1 IS NOW BEING LAUNCHED

MOVEMENT NOT DESIRED ENTER N(ONE) OTHERWISE ENTER SPEED
DESTINATION HEX  IN THIS FASHION

> 12
  > 1720

ENTER SEARCH MISSIONS AS FOLLOWS:
  0  NONE SCHEDULED
  1  ACTIVE MISSION
  2  PASSIVE MISSION

> 0

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
NONE SCHEDULED
DESTINATION 1720 SPEED 12
THERE IS NO SEARCH MISSION SCHEDULED

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION

> n

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

> 5

THIS OPTION WILL BEGIN THE NEXT GAME TURN. TYPE Y(ES)
IF YOU HAVE ENTERED ALL YOUR COMBAT, MOVE AND SEARCH
INSTRUCTIONS AND YOU ARE READY TO COMMENCE PLAY. TYPE N(O)
TO RETURN TO DISPLAY SELECTION.

> y

RED CALL TO STATUS ROUTINES. ENTER THE RED PASSWORD.

> redpas

THE FOLLOWING DISPLAY OPTIONS ARE AVAILABLE TO YOU:

1. DETECTIONS OF ENEMY PLATFORMS.
2. RESULTS OF WEAPONS FIRED.
3. DAMAGE TO FRIENDLY PLATFORMS.
4. STATUS OF FRIENDLY PLATFORMS AND INPUT OF INSTRUCTIONS.
5. BEGIN NEXT GAME TURN.

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

> 4

YOU HAVE SELECTED INDIVIDUAL PLATFORM STATUS DISPLAY AND
ASSOCIATED COMBAT, MOVEMENT AND SEARCH COMMAND INPUTS.
ENTER THE NAME OF THE DESIRED PLATFORM.

> ivan
IVAN IS LOCATED AT POSITION 1510 SPEED 0
IT WAS NOT ASSIGNED A SEARCH MISSION
IVAN WEAPON SYSTEM(S)
   TYPE  RANGE
   1 SSN-7  30
   2 21"  15

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

> n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

> 1
> 1511

ENTER SEARCH MISSIONS AS FOLLOWS:
   0 NONE SCHEDULED
   1 ACTIVE MISSION
   2 PASSIVE MISSION
   3 ACTIVE AND PASSIVE

> 2

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
   NONE SCHEDULED
DESTINATION 1511 SPEED 1
PASSIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> y

ENTER THE NAME OF THE DESIRED PLATFORM

> boris

162
BORIS IS LOCATED AT POSITION 1520 SPEED 0
IT WAS NOT ASSIGNED A SEARCH MISSION
BORIS WEAPON SYSTEM(S)
<table>
<thead>
<tr>
<th>TYPE</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21&quot;</td>
</tr>
</tbody>
</table>

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

> n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

> 2
> 1522

ENTER SEARCH MISSIONS AS FOLLOWS:
0  NONE SCHEDULED
1  ACTIVE MISSION
2  PASSIVE MISSION
3  ACTIVE AND PASSIVE

> 1

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
  NONE SCHEDULED
DESTINATION 1522  SPEED 2
ACTIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION.
> n

SELECT AND ENTER THE NUMBER FOR A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.
> 5
THIS OPTION WILL BEGIN THE NEXT GAME TURN. TYPE Y(ES)
IF YOU HAVE ENTERED ALL YOUR COMBAT, MOVE AND SEARCH
INSTRUCTIONS AND YOU ARE READY TO COMENCE PLAY. TYPE N(O)
TO RETURN TO DISPLAY SELECTION.

> y

BLUE CALL TO STATUS ROUTINES. ENTER THE BLUE PASSWORD.

> blupas

THE FOLLOWING DISPLAY OPTIONS ARE AVAILABLE TO YOU:

1. DETECTIONS OF ENEMY PLATFORMS.
2. RESULTS OF WEAPONS FIRED.
3. DAMAGE TO FRIENDLY PLATFORMS.
4. STATUS OF FRIENDLY PLATFORMS AND INPUT OF INSTRUCTIONS.
5. BEGIN NEXT GAME TURN.

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

> 1

<table>
<thead>
<tr>
<th>DETECTING PLATFORM</th>
<th>SONOR TYPE</th>
<th>SHIP TYPE</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TINCAN</td>
<td>ACTIVE</td>
<td>SUBMARINE</td>
<td>512</td>
</tr>
<tr>
<td>ESCORT</td>
<td>ACTIVE</td>
<td>SUBMARINE</td>
<td>512</td>
</tr>
<tr>
<td>ESCORT</td>
<td>ACTIVE</td>
<td>SUBMARINE</td>
<td>1522</td>
</tr>
<tr>
<td>TINCAN</td>
<td>PASSIVE</td>
<td>SUBMARINE</td>
<td>1522</td>
</tr>
<tr>
<td>LAPON</td>
<td>PASSIVE</td>
<td>SURFACE</td>
<td>502</td>
</tr>
<tr>
<td>LAPON</td>
<td>PASSIVE</td>
<td>SURFACE</td>
<td>720</td>
</tr>
<tr>
<td>LAPON</td>
<td>PASSIVE</td>
<td>SUBMARINE</td>
<td>1511</td>
</tr>
<tr>
<td>LAPON</td>
<td>PASSIVE</td>
<td>SUBMARINE</td>
<td>1522</td>
</tr>
</tbody>
</table>

END OF DETECTION ROUTINE. TYPE Y(ES) TO CONTINUE.

> y

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

> 4
YOU HAVE SELECTED INDIVIDUAL PLATFORM STATUS DISPLAY AND ASSOCIATED COMBAT, MOVEMENT AND SEARCH COMMAND INPUTS. ENTER THE NAME OF THE DESIRED PLATFORM.

> tincan

TINCAN IS LOCATED AT POSITION 502 SPEED 2
IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)
ACTIVE AND PASSIVE

<table>
<thead>
<tr>
<th>TYPE</th>
<th>LOCATION</th>
<th>METHOD</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBMARINE</td>
<td>512</td>
<td>ACTIVE</td>
<td>10</td>
</tr>
<tr>
<td>SUBMARINE</td>
<td>1522</td>
<td>PASSIVE</td>
<td>25</td>
</tr>
</tbody>
</table>

TINCAN WEAPON SYSTEM(S)

<table>
<thead>
<tr>
<th>TYPE</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MK 46</td>
</tr>
<tr>
<td>2</td>
<td>ASROC</td>
</tr>
</tbody>
</table>

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

> n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

> 3
> 802

ENTER SEARCH MISSIONS AS FOLLOWS:
0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION
3 ACTIVE AND PASSIVE

> 3

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
NONE SCHEDULED
DESTINATION 802 SPEED 3
ACTIVE AND PASSIVE SEARCH MISSION

165
ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(0) TO RETURN TO DISPLAY SELECTION
> y

ENTER THE NAME OF THE DESIRED PLATFORM
> escort

ESCORT IS LOCATED AT POSITION 720 SPEED 2
IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)
ACTIVE

<table>
<thead>
<tr>
<th>TYPE</th>
<th>LOCATION</th>
<th>METHOD</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBMARINE</td>
<td>512</td>
<td>ACTIVE</td>
<td>9</td>
</tr>
<tr>
<td>SUBMARINE</td>
<td>1522</td>
<td>ACTIVE</td>
<td>8</td>
</tr>
</tbody>
</table>

ESCORT WEAPON SYSTEM(S)

<table>
<thead>
<tr>
<th>TYPE</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ASROC 7</td>
</tr>
<tr>
<td>2</td>
<td>MK 46 20</td>
</tr>
</tbody>
</table>

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

>1522
>1
TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER
OTHERWISE ENTER N(0)

>2
TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER
OTHERWISE ENTER N(0)

>n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

>2
>72

YOU HAVE EITHER SET A SPEED HIGHER THAN MAX CAPABLE OR
ATTEMPTED TO GO FURTHER THAN THE SPEED SETTING WILL ALLOW

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

>2
>1722
YOU HAVE EITHER SET A SPEED HIGHER THAN MAX CAPABLE OR ATTEMPTED TO GO FURTHER THAN THE SPEED SETTING WILL ALLOW

TO SCHEDULE MOVEMENT ENTER SPEED DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

> 2
> 722

ENTER SEARCH MISSIONS AS FOLLOWS:
0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION
3 ACTIVE AND PASSIVE

> 3

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
ASROC WEAPON TGT LOCATION 1522
MK 46 WEAPON TGT LOCATION 1522
DESTINATION 722 SPEED 2
ACTIVE AND PASSIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> y

ENTER THE NAME OF THE DESIRED PLATFORM

> lapon

LAPON IS LOCATED AT POSITION 512 SPEED 2
IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)

PASSIVE

<table>
<thead>
<tr>
<th>TYPE</th>
<th>LOCATION</th>
<th>METHOD</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURFACE</td>
<td>502</td>
<td>PASSIVE</td>
<td>10</td>
</tr>
<tr>
<td>SURFACE</td>
<td>720</td>
<td>PASSIVE</td>
<td>9</td>
</tr>
<tr>
<td>SUBMARINE</td>
<td>1511</td>
<td>PASSIVE</td>
<td>10</td>
</tr>
<tr>
<td>SUBMARINE</td>
<td>1522</td>
<td>PASSIVE</td>
<td>15</td>
</tr>
</tbody>
</table>
LAPON WEAPON SYSTEM(S)

<table>
<thead>
<tr>
<th>TYPE</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROC 25</td>
</tr>
<tr>
<td>2</td>
<td>MK 48 25</td>
</tr>
</tbody>
</table>

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER TARGET LOCATION WEAPON SYSTEM NUMBER IN THIS FASHION

> 1522
> 1

TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER OTHERWISE ENTER N(O)

> 2

TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER OTHERWISE ENTER N(O)

> n

TO SCHEDULE MOVEMENT ENTER SPEED DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

> 2
> 712

ENTER SEARCH MISSIONS AS FOLLOWS:
0  NONE SCHEDULED
1  ACTIVE MISSION
2  PASSIVE MISSION
3  ACTIVE AND PASSIVE

> 1

YOU HAVE ENTERED THE FOLLOWING COMMANDS

COMBAT:
SUBROC  WEAPON TGT LOCATION 1522
MK 48  WEAPON TGT LOCATION 1522
DESTINATION 712  SPEED 2
ACTIVE SEARCH MISSION
ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> Y

ENTER THE NAME OF THE DESIRED PLATFORM

> spru1

SPRU1 IS CURRENTLY NOT FLYING - IT IS ASSUMED YOU DESIRE TO LAUNCH.

SPRU1 IS NOW BEING LAUNCHED

MOVEMENT NOT DESIRED ENTER N(ONE) OTHERWISE ENTER SPEED DESTINATION HEX IN THIS FASHION

> 10
> 1502

ENTER SEARCH MISSIONS AS FOLLOWS:
0  NONE SCHEDULED
1  ACTIVE MISSION
2  PASSIVE MISSION

> 0
YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
NONE SCHEDULED
DESTINATION 1502 SPEED 10
THERE IS NO SEARCH MISSION SCHEDULED

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> Y

ENTER THE NAME OF THE DESIRED PLATFORM

> knox1

169
KNOX1 IS LOCATED AT POSITION 1720 AT SPEED 12
IT WAS NOT ASSIGNED A SEARCH MISSION

NO COMBAT ENTER N(ONE) OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

> n
MOVEMENT NOT DESIRED ENTER N(ONE) OTHERWISE ENTER
SPEED
DESTINATION HEX IN THIS FASHION

> n

ENTER SEARCH MISSIONS AS FOLLOWS:
0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION

> 2
YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
NONE SCHEDULED
SPEED IS ZERO AT LOCATION 1720
A PASSIVE SEARCH MISSION WILL BE CONDUCTED

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION

> n

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

> 5

THIS OPTION WILL BEGIN THE NEXT GAME TURN. TYPE Y(ES)
IF YOU HAVE ENTERED ALL YOUR COMBAT, MOVE AND SEARCH
INSTRUCTIONS AND YOU ARE READY TO COMMENCE PLAY. TYPE N(O)
TO RETURN TO DISPLAY SELECTION.

> y

RED CALL TO STATUS ROUTINES. ENTER THE RED PASSWORD.

> redpas

170
THE FOLLOWING DISPLAY OPTIONS ARE AVAILABLE TO YOU:

1. DETECTIONS OF ENEMY PLATFORMS.
2. RESULTS OF WEAPONS FIRED.
3. DAMAGE TO FRIENDLY PLATFORMS.
4. STATUS OF FRIENDLY PLATFORMS AND INPUT OF INSTRUCTIONS.
5. BEGIN NEXT GAME TURN.

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

> 1

<table>
<thead>
<tr>
<th>DETECTING PLATFORM</th>
<th>SONOR TYPE</th>
<th>SHIP TYPE</th>
<th>DETECTED LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVAN</td>
<td>PASSIVE</td>
<td>SURFACE</td>
<td>502</td>
</tr>
<tr>
<td>IVAN</td>
<td>PASSIVE</td>
<td>SURFACE</td>
<td>720</td>
</tr>
<tr>
<td>IVAN</td>
<td>PASSIVE</td>
<td>SUBMARINE</td>
<td>1522</td>
</tr>
</tbody>
</table>

END OF DETECTION ROUTINE. TYPE Y(ES) TO CONTINUE.

> y

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

> 4

YOU HAVE SELECTED INDIVIDUAL PLATFORM STATUS DISPLAY AND ASSOCIATED COMBAT, MOVEMENT AND SEARCH COMMAND INPUTS.
ENTER THE NAME OF THE DESIRED PLATFORM.

> ivan

IVAN IS LOCATED AT POSITION 1511 SPEED 1
IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)

<table>
<thead>
<tr>
<th>PASSIVE TYPE</th>
<th>LOCATION</th>
<th>METHOD</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURFACE</td>
<td>502</td>
<td>PASSIVE</td>
<td>14</td>
</tr>
<tr>
<td>SURFACE</td>
<td>720</td>
<td>PASSIVE</td>
<td>13</td>
</tr>
<tr>
<td>SUBMARINE</td>
<td>1522</td>
<td>PASSIVE</td>
<td>11</td>
</tr>
</tbody>
</table>
IVAN WEAPON SYSTEM(S)

<table>
<thead>
<tr>
<th>TYPE</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SSN-7</td>
</tr>
<tr>
<td>2</td>
<td>21&quot;</td>
</tr>
</tbody>
</table>

NO COMBAT ENTER N(ONE) - OTHERWISE
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

>502
>1
TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER
OTHERWISE ENTER N(O)

>2
TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER
OTHERWISE ENTER N(O)

>n

TO SCHEDULE MOVEMENT ENTER SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

>2
>1311

ENTER SEARCH MISSIONS AS FOLLOWS:
0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION
3 ACTIVE AND PASSIVE

>1

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
SSN-7 WEAPON TGT LOCATION 502
21" WEAPON TGT LOCATION 502
DESTINATION 1311 SPEED 2
ACTIVE SEARCH MISSION
ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
\> y

ENTER THE NAME OF THE DESIRED PLATFORM

\> boris

BORIS IS LOCATED AT POSITION 1522 SPEED 2
IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)
ACTIVE
TYPE LOCATION METHOD RANGE

THERE WERE NO DETECTIONS
BORIS WEAPON SYSTEM(S)
TYPE RANGE
1 21" 15

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

\> n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

\> 2
\> 1722

ENTER SEARCH MISSIONS AS FOLLOWS:
0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION
3 ACTIVE AND PASSIVE

\> 1

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
NONE SCHEDULED

173
DESTINATION 1722  SPEED 2
ACTIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> n

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS
> y

YOU HAVE MADE AN ERROR IN YOUR INPUT SELECTION. TRY AGAIN.

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.
> 5

THIS OPTION WILL BEGIN THE NEXT GAME TURN. TYPE Y(ES) IF YOU HAVE ENTERED ALL YOUR COMBAT, MOVE AND SEARCH INSTRUCTIONS AND YOU ARE READY TO COMMENCE PLAY. TYPE N(O) TO RETURN TO DISPLAY SELECTION.
> y

THE ATTACK BY ESCORT FAILED AS TGT RANGE > WEAPON RANGE

YOU HAVE JUST COMPLETED GAME TURN 2. TYPE Y(ES) IF YOU WANT TO CONTINUE THE GAME, OR N(O) TO ALLOW THE GAME TO END.
> y

BLUE CALL TO STATUS ROUTINES. ENTER THE BLUE PASSWORD.
> blupas

THE FOLLOWING DISPLAY OPTIONS ARE AVAILABLE TO YOU:

1. DETECTIONS OF ENEMY PLATFORMS.

2. RESULTS OF WEAPONS FIRED.

3. DAMAGE TO FRIENDLY PLATFORMS.

4. STATUS OF FRIENDLY PLATFORMS AND INPUT OF INSTRUCTIONS.

5. BEGIN NEXT GAME TURN.
SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

>3

SHIP STATUS/HITS

TINCAN 0

ESCORT 0

LAPON 0

END OF DAMAGE DISPLAY ROUTINE. TYPE Y(ES) TO CONTINUE.

> y

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

>2

TYPE Y(ES) FOR SHIP RESULTS.

> y

SHIP - ESCORT  COMBAT LOCATION  - 1522
WEAPON 1  TYPE  -  ASROC  PROB(HIT)/DETONATION  -  NO DETONATION
WEAPON 2  TYPE  -  MK 46  PROB(HIT)/DETONATION  -  DETONATION

SHIP - LAPON  COMBAT LOCATION  - 1522
WEAPON 1  TYPE  -  SUBROC  PROB(HIT)/DETONATION  - 0.6667
WEAPON 2  TYPE  -  MK 48  PROB(HIT)/DETONATION  - DETONATION

END OF WEAPON RESULT DISPLAY. TYPE Y(ES) TO CONTINUE.

> y

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.

>1

SONOR  DETECTING  SHIP  TYPE  TYPE  PLATFORM  USED  DETECTED  LOCATION

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ESCORT</td>
<td>ACTIVE</td>
<td>SUBMARINE</td>
<td></td>
<td></td>
<td>712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNOX1</td>
<td>PASSIVE</td>
<td>SUBMARINE</td>
<td></td>
<td></td>
<td>712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNOX1</td>
<td>PASSIVE</td>
<td>SUBMARINE</td>
<td></td>
<td></td>
<td>1311</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

175
SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.
> 4

YOU HAVE SELECTED INDIVIDUAL PLATFORM STATUS DISPLAY AND 
ASSOCIATED COMBAT, MOVEMENT AND SEARCH COMMAND INPUTS.
ENTER THE NAME OF THE DESIRED PLATFORM.

>tincan

TINCAN IS LOCATED AT POSITION 802 SPEED 3
IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)
ACTIVE AND PASSIVE
TYPE LOCATION METHOD RANGE
SUBMARINE 712 PASSIVE 10
SUBMARINE 1311 PASSIVE 11
TINCAN WEAPON SYSTEM(S)
TYPE RANGE
1 MK 46 20
2 ASROC 7

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

>1311
>1

TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER
OTHERWISE ENTER N(O)

>n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)
ENTER SEARCH MISSIONS AS FOLLOWS:
0  NONE SCHEDULED
1  ACTIVE MISSION
2  PASSIVE MISSION
3  ACTIVE AND PASSIVE

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
MK 46  WEAPON  TGT LOCATION 1311
DESTINATION 1102  SPEED 3
ACTIVE AND PASSIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION

ENTER THE NAME OF THE DESIRED PLATFORM

escort

ESCORT IS LOCATED AT POSITION  722  SPEED 2
THE ESCORT ATTACK ON TARGET LOCATED AT 1522
RELEASE OF ASROC  RESULTED IN
NO DETONATION - NO DISTURBED WATER MARKER
RELEASE OF MK 46  RESULTED IN
A DETONATION(HIT)

IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)
ACTIVE AND PASSIVE

<table>
<thead>
<tr>
<th>TYPE</th>
<th>LOCATION</th>
<th>METHOD</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBMARINE</td>
<td>712</td>
<td>ACTIVE</td>
<td>10</td>
</tr>
<tr>
<td>SUBMARINE</td>
<td>712</td>
<td>PASSIVE</td>
<td>10</td>
</tr>
<tr>
<td>SUBMARINE</td>
<td>1311</td>
<td>PASSIVE</td>
<td>14</td>
</tr>
</tbody>
</table>

ESCORT WEAPON SYSTEM(S)

<table>
<thead>
<tr>
<th>TYPE</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASROC</td>
<td>7</td>
</tr>
<tr>
<td>MK 46</td>
<td>20</td>
</tr>
</tbody>
</table>

177
NO COMBAT ENTER N(ONE) - OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

>1311
>2
TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER
OTHERWISE ENTER N(0)

>n

TO SCHEDULE MOVEMENT ENTER
SPEED
DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

>2
>720

ENTER SEARCH MISSIONS AS FOLLOWS:
0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION
3 ACTIVE AND PASSIVE

>3

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
M K 46 WEAPON TGT LOCATION 1311
DESTINATION 720 SPEED 2
ACTIVE AND PASSIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(0) TO RETURN TO DISPLAY SELECTION
>y

ENTER THE NAME OF THE DESIRED PLATFORM

>lapon

LAPON IS LOCATED AT POSITION 712 SPEED 2

178
THE LAPON ATTACK ON TARGET LOCATED AT 1522

RELEASE OF SUBROC RESULTED IN A DETONATION WITH PROBABILITY OF A HIT OF 4 OUT OF 6

RELEASE OF MK 48 RESULTED IN A DETONATION (HIT)

IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)

<table>
<thead>
<tr>
<th>ACTIVE</th>
<th>Type</th>
<th>Location</th>
<th>Method</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SUBMARINE</td>
<td>1311</td>
<td>ACTIVE</td>
<td>6</td>
</tr>
<tr>
<td>LAPON WEAPON SYSTEM(S)</td>
<td>TYPE</td>
<td>RANGE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SUBROC</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MK 48</td>
<td>25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER TARGET LOCATION WEAPON SYSTEM NUMBER IN THIS FASHION

> 1311
> 2

TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER OTHERWISE ENTER N(0)

> n

TO SCHEDULE MOVEMENT ENTER SPEED DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

> 2
> 214

ENTER SEARCH MISSIONS AS FOLLOWS:

0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION
3 ACTIVE AND PASSIVE

> 1
YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
MK 48 WEAPON TGT LOCATION 1311
DESIgnATION 714 SPEED 2
ACTIVE SEARCH MISSION

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> y

ENTER THE NAME OF THE DESIRED PLATFORM

> spru1

SPRU1 IS LOCATED AT POSITION 1502 AT SPEED 10
IT WAS NOT ASSIGNED A SEARCH MISSION

NO COMBAT ENTER N(ONE) OTHERWISE ENTER
TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

> n
MOVEMENT NOT DESIRED ENTER N(ONE) OTHERWISE ENTER
SPEED
DESTINATION HEX IN THIS FASHION

> n

ENTER SEARCH MISSIONS AS FOLLOWS:
 0 NONE SCHEDULED
 1 ACTIVE MISSION
 2 PASSIVE MISSION

> 2

YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
NONE SCHEDULED
SPEED IS ZERO AT LOCATION 1502
A PASSIVE SEARCH MISSION WILL BE CONDUCTED
ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION

> y

ENTER THE NAME OF THE DESIRED PLATFORM

> knox11
YOU HAVE ENTERED AN INCORRECT PLATFORM NAME.
REENTER

> knox1

KNOX1 IS LOCATED AT POSITION 1720 AT SPEED 0
IT CONDUCTED A PASSIVE SEARCH WITH FOLLOWING RESULTS
SUBMARINE CONTACT LOCATION RANGE
    712      13
    1311     11
KNOX1 WEAPON SYSTEM(S)
    TYPE RANGE
    1      MK 46    20

NO COMBAT ENTER N(ONE) OTHERWISE ENTER
TARGET LOCATION WEAPON SYSTEM NUMBER IN THIS FASHION

> 1311
> 1
MOVEMENT NOT DESIRED ENTER N(ONE) OTHERWISE ENTER
SPEED DESTINATION HEX IN THIS FASHION

> n

ENTER SEARCH MISSIONS AS FOLLOWS:
0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION

> 1
YOU HAVE ENTERED THE FOLLOWING COMMANDS
COMBAT:
MK 46 WEAPON TGT LOCATION 1311
SPEED IS ZERO AT LOCATION 1720
AN ACTIVE SEARCH MISSION WILL BE CONDUCTED

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> n

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.
> 5

THIS OPTION WILL BEGIN THE NEXT GAME TURN. TYPE Y(ES)
IF YOU HAVE ENTERED ALL YOUR COMBAT, MOVE AND SEARCH
INSTRUCTIONS AND YOU ARE READY TO COMMENCE PLAY. TYPE N(O)
TO RETURN TO DISPLAY SELECTION.
> y

RED CALL TO STATUS ROUTINES. ENTER THE RED PASSWORD.
> redpas

THE FOLLOWING DISPLAY OPTIONS ARE AVAILABLE TO YOU:

1. DETECTIONS OF ENEMY PLATFORMS.
2. RESULTS OF WEAPONS FIRED.
3. DAMAGE TO FRIENDLY PLATFORMS.
4. STATUS OF FRIENDLY PLATFORMS AND INPUT OF INSTRUCTIONS.
5. BEGIN NEXT GAME TURN.

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.
> 3

<table>
<thead>
<tr>
<th>SHIP</th>
<th>STATUS/HITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVAN</td>
<td>0</td>
</tr>
<tr>
<td>BORIS</td>
<td>SUNK</td>
</tr>
</tbody>
</table>

END OF DAMAGE DISPLAY ROUTINE. TYPE Y(ES) TO CONTINUE.
> y

182
SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.
> 2

SHIP - IVAN    COMBAT LOCATION - 502
WEAPON 1 TYPE - SSN-7    PROB(HIT)/DETONATION - 0.1607
WEAPON 2 TYPE - 21"    PROB(HIT)/DETONATION-NO DETONATION

END OF WEAPON RESULT DISPLAY. TYPE Y(ES) TO CONTINUE.
>y

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.
> 1

DETECTING   SONOR   SHIP
PLATFORM     TYPE   TYPE
           USED   DETECTED   LOCATION
IVAN     ACTIVE   SUBMARINE   712

END OF DETECTION ROUTINE. TYPE Y(ES) TO CONTINUE.
>y

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.
> 4

YOU HAVE SELECTED INDIVIDUAL PLATFORM STATUS DISPLAY AND
ASSOCIATED COMBAT, MOVEMENT AND SEARCH COMMAND INPUTS.
ENTER THE NAME OF THE DESIRED PLATFORM.

> ivan

IVAN IS LOCATED AT POSITION 1311 SPEED 2

THE IVAN ATTACK ON TARGET LOCATED AT 502

RELEASE OF SSN-7 RESULTED IN
A DETONATION WITH PROBABILITY
OF A HIT OF 1 OUT OF 6
RELEASE OF 21" RESULTED IN NO DETONATION - NO DISTURBED WATER MARKER

IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)

ACTIVE

<table>
<thead>
<tr>
<th>TYPE</th>
<th>LOCATION</th>
<th>METHOD</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBMARINE 712</td>
<td>ACTIVE</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>IVAN WEAPON SYSTEM(S)</td>
<td>TYPE</td>
<td>RANGE</td>
<td></td>
</tr>
<tr>
<td>1 SSN-7</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 21&quot;</td>
<td>15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NO COMBAT ENTER N(ONE) - OTHERWISE ENTER TARGET LOCATION
WEAPON SYSTEM NUMBER IN THIS FASHION

> 712
> 2

TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER OTHERWISE ENTER N(O)

> n

TO SCHEDULE MOVEMENT ENTER SPEED DESTINATION IN THAT ORDER, OTHERWISE ENTER N(ONE)

> 2
> 1111

ENTER SEARCH MISSIONS AS FOLLOWS:

0 NONE SCHEDULED
1 ACTIVE MISSION
2 PASSIVE MISSION
3 ACTIVE AND PASSIVE

> 1

YOU HAVE ENTERED THE FOLLOWING COMMANDS COMBAT:

21" WEAPON TGT LOCATION 712 DESIGNATION 1111 SPEED 2
ACTIVE SEARCH MISSION
ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> y

ENTER THE NAME OF THE DESIRED PLATFORM

> boris

BORIS IS LOCATED AT POSITION 1522 SPEED 2

NOTE
DUE TO PREVIOUS AND/OR CURRENT DAMAGE SUFFERED, MAX SPEED IS NOW 0 THE SHIP HAS BEEN HIT 3 TIMES

IT HAS BEEN HIT BY A TORPEDO
IT HAS BEEN HIT BY A NON-TORPEDO WEAPON
IT HAS BEEN HIT BY A TORPEDO
IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S)
ACTIVE
<table>
<thead>
<tr>
<th>TYPE</th>
<th>LOCATION</th>
<th>METHOD</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
THERE WERE NO DETECTIONS

SINCE THE BORIS HAS BEEN SUNK, NO COMMAND INPUTS CAN BE ACCEPTED

ENTER Y(ES) TO CONTINUE WITH ANOTHER PLATFORM
ENTER N(O) TO RETURN TO DISPLAY SELECTION
> n

SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
ENTER A 6 FOR A LIST OF DISPLAY OPTIONS
> 5

THIS OPTION WILL BEGIN THE NEXT GAME TURN. TYPE Y(ES)
IF YOU HAVE ENTERED ALL YOUR COMBAT, MOVE AND SEARCH INSTRUCTIONS AND YOU ARE READY TO COMMENCE PLAY. TYPE N(O) TO RETURN TO DISPLAY SELECTION.
> y
BLUE RESULTS:

<table>
<thead>
<tr>
<th>SHIP</th>
<th>STATUS/HITS</th>
<th>FINAL LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TINCan</td>
<td>0</td>
<td>1102</td>
</tr>
<tr>
<td>ESCOrt</td>
<td>0</td>
<td>720</td>
</tr>
<tr>
<td>LAPOn</td>
<td>0</td>
<td>714</td>
</tr>
</tbody>
</table>

TYPE Y(ES) TO SEE RED RESULTS. 
＞y

RED RESULTS:

<table>
<thead>
<tr>
<th>SHIP</th>
<th>STATUS/HITS</th>
<th>FINAL LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVAN</td>
<td>SUNK</td>
<td>1311</td>
</tr>
<tr>
<td>BORIS</td>
<td>SUNK</td>
<td>1522</td>
</tr>
</tbody>
</table>

TYPE Y(ES) TO RETURN TO MENU SELECTION AND START A NEW GAME. TYPE N(O) TO EXIT THE GAME. 
＞y

THE FOLLOWING OPTIONS ARE AVAILABLE TO YOU:

1. ZERO ANY PREVIOUSLY ESTABLISHED DATA BASE.
2. EXIT THE GAME.
3. INITIALIZE GAME PARAMETERS AND COMMENCE PLAY.

ENTER AN OPTION.
＞2

EXISTING THE GAME.
R;

186
APPENDIX F
PROGRAM LISTING

1 IMPLICIT INTEGER (A-Z)
2 DIMENSION SHIPOA(130, 181), NAMESH(ISO, 81), ACATA(16, 7), NAMEAC(16, 7)
3 155(130, 56), SHIPOA(130, 57), SHIPOA(130, 58), SHIPOA(130, 59), SHIPOA(130, 60),
4 151(130, 51), SHIPOA(130, 51), SHIPOA(130, 51), SHIPOA(130, 51),
5 1S1(130), SI(130), S14(130), SI(130), SI(130), SI(130), S17(130),
6 SI(130), SI(130), SI(130), SI(130), S14(130), S14(130), S14(130),
7 1NS(130), 1NS(130), 1NS(130),
8 INPUT(10), ALPHA(2), CHECK(6), CLSRNA(8)
9 EQUIVALENCE (SHIPDA(1, 1), (SHIPDA(1, 2), S2(1))
10 (SHIPDA(1, 3), S3(1)), (SHIPDA(1, 4), S4(1)), (SHIPDA(1, 5), S5(1)),
11 (SHIPDA(1, 6), S6(1)), (SHIPDA(1, 7), S7(1)), (SHIPDA(1, 8), S8(1)),
12 (SHIPDA(1, 9), S9(1)), (SHIPDA(1, 10), S10(1)), (SHIPDA(1, 11), S11(1)),
13 (SHIPDA(1, 12), S12(1)), (SHIPDA(1, 13), S13(1)), (SHIPDA(1, 14), S14(1)),
14 (SHIPDA(1, 15), S15(1)), (SHIPDA(1, 16), S16(1)), (SHIPDA(1, 17), S17(1)),
15 (SHIPDA(1, 18), S18(1)), (NAMEAC(1, 1), NAMEAC(1, 1)), (NAMEAC(1, 2), NAMEAC(1, 2)),
16 (NAMEAC(1, 3), NAMEAC(1, 3)), (NAMEAC(1, 4), NAMEAC(1, 4)), (NAMEAC(1, 5), NAMEAC(1, 5)),
17 (NAMEAC(1, 6), NAMEAC(1, 6)), (NAMEAC(1, 7), NAMEAC(1, 7)), (NAMEAC(1, 8), NAMEAC(1, 8))
18 C COMMON STATEMENT
19 C COMMON PLATFORM(100, 37), DETECT(100, 4), WEP(29, 3), HIT(100, 3), SUNK(100)
20 C 1, WATER(100), NAME(100, 6), BUFFER(100, 8), RUDAP(6), REDAP(6),
21 21 PLAT, AIR, ISURF, ISUB, ADV, PVGQ, PEY, ERATIC, BLURAIR, REDAIR, BLUSUR,
22 3RESUR, BLUSUB, REDSUB, PBAIR1, PBAIR2, PBAIR1, PBAIR2, PBAIR1, PBAIR2,
23 WPUR1, WPUR1, WPUR2, WPUR2, WPUR1, WPUR2
24 C AIRCRAFT NAME DATA ARRAY
26 C "ML", "BA", "SA", "BA", "MA", "ML"
28 C 1"T", "IL", "Y", "AC", "10", "KI", "HR", "IZ", "AG", "HI", "2W", "5P",
31 C 1, "EL", "8W", "5W", "11", "11", "ON", "8W"
32 C AIRCRAFT DATA ARRAY
33 C DATA ACDATA/11, 12, 15, 10, 12, 16, 12, 14, 10, 24, 31, 15, 30, 2, 35, 24, 3, 4, 3,
34 C 12, 4, 3, 3, 4, 4, 3, 3, 2, 4, 3, 4, 3, 4, 3, 4, 4, 4, 3, 4, 3, 4, 3, 4, 3, 2, 4,
35 C 12, 29, 4, 16, 17, 9, 15, 4, 5, 13, 12, 3, 2, 29, 9, 0, 5, 29, 3, 3, 9, 0, 3, 4, 3, 2, 4,
36 C WEAPON NAME DATA ARRAY
41 C 1", "3", "1", "OC", "14", "2W", "8", "7C", "AF", "0", "2", "5",

187
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>151</td>
<td>DATA</td>
<td>57/6=2.4=3.2.4=3.1.2=3.2=2.3.7=2.3.3=2.5.3.2=2.4=3.5=2.2=1.</td>
</tr>
<tr>
<td>152</td>
<td></td>
<td>12.1=2.0.2.2.3.2=2.17=2.6=2.1.2.1.2.2.2=1.6.3=2.3.4=2.3=3.</td>
</tr>
<tr>
<td>153</td>
<td></td>
<td>12=0.2=2.3=2.1.5=2.0.2.2=1.3=2.3=1.</td>
</tr>
<tr>
<td>154</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>155</td>
<td></td>
<td>129.3=21.4=5.2.5.1.15.0.10.3=16.7=23.5=9.5=16.9.0.18.23.16.18.9=30.</td>
</tr>
<tr>
<td>156</td>
<td></td>
<td>15=12.4=11.12.3=11.5=13.11=17.3=14.17.3=14.2=8.</td>
</tr>
<tr>
<td>157</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>158</td>
<td></td>
<td>14=7.6.4=4.1.4.1=0.3.3=4.7=6.5=4.6=4.0.3=4.9.3=0.5=5.8=4.5=5.</td>
</tr>
<tr>
<td>159</td>
<td></td>
<td>11=2.5.4.5.3=4.2=5/7.</td>
</tr>
<tr>
<td>160</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>161</td>
<td></td>
<td>16.0=25.4=0.5.0.5.0.0.23.3=0.16.5=0.5=10.0.4=23.10.4=23.10.5=0.</td>
</tr>
<tr>
<td>162</td>
<td></td>
<td>14=19.2=25.19.25.2=26.25.5=0.4=0.27.0.2=27.0.27.24.3=0.15.3=28.2=0/</td>
</tr>
<tr>
<td>163</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>164</td>
<td></td>
<td>10.4=0.4.0.4.0.0.8.3=0.4.6=0.5=4.0.2=4.5.4=0.2=4.5.4=0.4=2.8=5.</td>
</tr>
<tr>
<td>165</td>
<td></td>
<td>14.4=6.5=4.4=0.6.0.5=6.0.2=6.3=0.5=6.0=0/</td>
</tr>
<tr>
<td>166</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>167</td>
<td></td>
<td>129.6=0.14=0.6=0/</td>
</tr>
<tr>
<td>168</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>169</td>
<td></td>
<td>11=0.6=0/</td>
</tr>
<tr>
<td>170</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>171</td>
<td></td>
<td>19=0.2=2.4=0.2=2.0=2.0=16.8=0.3=3.2=0.3=12=0.2=9.2=9.2=0.9=0.</td>
</tr>
<tr>
<td>172</td>
<td></td>
<td>16=0/</td>
</tr>
<tr>
<td>173</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>174</td>
<td></td>
<td>110.1.2.5=1.9=0.2=2.1.3=0.5.2=0.1.2=0.10.6.0.3=5.2=0.1.2=0.</td>
</tr>
<tr>
<td>175</td>
<td></td>
<td>11=0.9.4.1.1=0.2=2.1=1/</td>
</tr>
<tr>
<td>176</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>177</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>178</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>179</td>
<td></td>
<td>14=1.2.3=2.13=1.5=2.2=2.9=1.3=2.4=1.2=2/</td>
</tr>
<tr>
<td>180</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>181</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>182</td>
<td></td>
<td>DATA</td>
</tr>
<tr>
<td>183</td>
<td></td>
<td>DD</td>
</tr>
<tr>
<td>184</td>
<td></td>
<td>DD</td>
</tr>
<tr>
<td>185</td>
<td></td>
<td>WEP</td>
</tr>
<tr>
<td>186</td>
<td></td>
<td>CONTINUE</td>
</tr>
</tbody>
</table>

190
201 CONTINUE
202 C
203 C INITIALIZE THE GAME PARAMETERS
204 C
205 TURN=0
206 BLAIR=0
207 BLUSUR=0
208 BLUSUB=0
209 REDAIR=0
210 REDSUR=0
211 REDSUB=0
212 C
213 C INITIALIZE THE ARRAY PLATFM AND OTHERS
214 C
215 DD B3 J=1,MAXPL
216 DD B2 J=1,37
217 PLATFM(I,J)=0
218 CONTINUE
219 WATER(I,J)=1
220 GETCLT(I,J,J)=1
221 HIT(I,J,J)=1
222 SUNH(I,J)=1
223 BUFFER(I,J)=0
224 BUFFER(I,J,J)=0
225 BUFFER(I,J,J,J)=0
226 BUFFER(I,J,J,J,J)=0
227 BUFFER(I,J,J,J,J,J)=0
228 BUFFER(I,J,J,J,J,J,J)=0
229 BUFFER(I,J,J,J,J,J,J,J)=0
230 BUFFER(I,J,J,J,J,J,J,J,J)=0
231 CONTINUE
232 C
233 C MENU
234 C
235 WRITE(6,100)
236 FORMAT(' THE FOLLOWING OPTIONS ARE AVAILABLE TO YOU:','
237 /*/,' 1. ZERO ANY PREVIOUSLY ESTABLISHED DATA BASE ','
238 /*/,' 2. EXIT THE GAME ','
239 /*/,' 3. INITIALIZE GAME PARAMETERS AND COMMENCE PLAY ','
240 /*/,' ENTER AN OPTION ')
241 READ(5,200) INPUT
242 FORMAT(10)
243 CALL NONVRT(INPUT,INDEX)
244 IF(INDEX,NE.1) GO TO 2
245 FORMAT(10)
246 C
247 WRITE(6,111)
248 FORMAT(' THE OPTION YOU HAVE SELECTED WILL DESTROY ANY FILES THAT ','
249 /*/,' YOU HAVE ALREADY CREATED. IF YOU CONTINUE, YOU WILL HAVE ','
250 /*/,' TO RELOAD ALL USER INPUTS. TYPE YES TO ERASE THE FILES ')
1/,' OR TYPE N(0) IF YOU WANT TO RETURN TO MENU SELECTION WITHOUT'.
252 1/,' ERASING THE FILES.'
253 READ(5,200) INPUT
254 CALL INVRT1(INPUT,A)
255 IF (A.NE.ALPHA(1)) GO TO 1
256 GO 0 99
257 2 IF (INDEX.EQ.2) GO TO 9999
258 C
259 C LOAD GAME PARAMETERS
260 C
261 4 HALF=MAXPLT
262 4 HALF=HALF/2
263 WRITE(6,120) MAXPLT,HALF
264 120 FORMAT(1,'YOU ARE NOW GOING TO ENTER THE BASIC GAME PARAMETERS.'),
265 1/,' ONE WORD OF CAUTION. THE PRESENT GAME CAN ONLY HANDLE A',
266 1/,' TOTAL OF ',14,' WEAPON SYSTEMS. INCLUDED IN THESE SYSTEMS',
267 1/,' ARE SURFACE SHIPS, SUBMARINES AND AIRCRAFT. THE NUMBER OF',
268 1/,' AIRCRAFT ARE DETERMINED BY THE SHIP CLASS AS REFLECTED IN THE',
269 1/,' DATA BASE, AND WILL BE LOADED AUTOMATICALLY WHEN YOU SELECT',
270 1/,' THE SHIP CLASS. PRIOR TO BEGINNING THE LOADING PROCEDURE',
271 1/,' YOU SHOULD ENSURE THAT THE TOTAL NUMBER OF EACH PLAYERS',
272 1/,' WEAPON SYSTEMS DOES NOT EXCEED ',14,'.'
273 WRITE(6,121)
274 121 FORMAT(/1,' ENTER THE NUMBER OF GAME TURNS YOU WANT TO PLAY.'),
275 1/2 READ(5,200) INPUT
276 CALL NONVRT(INPUT,TURN)
277 WRITE(6,167)
278 167 FORMAT(1,'ENTER THE WATER CONDITION (O-NORMAL, I-ERRATIC).')
279 READ(5,200) INPUT
280 CALL NONVRT(INPUT,ERATIC)
281 IF(ERATIC.EQ.0.OR.ERATIC.EQ.1) GO TO 5
282 WRITE(6,168)
283 168 FORMAT(1,' YOU HAVE MADE AN ERROR ENTERING THE WATER CONDITION.')
284 GO TO 168
285 5 WRITE(6,122)
286 122 FORMAT(1,' THE FOLLOWING INPUTS APPLY ONLY TO THE BLUE PLAYER',)
287 1/,' THE FIRST IS THE BLUE PLATER PASSWORD WHICH IS',
288 1/,' A MAXIMUM OF 6 CHARACTERS. THE PASSWORD IS IMPORTANT BECAUSE'
289 1/,' IT IS REQUIRED PRIOR TO THE INPUT OF COMBAT, MOVE AND SEARCH',
290 1/,' COMMANDS. IF YOU FORGET THE PASSWORD, YOU WILL HAVE TO RETUR
291 1/,' TO MENU SELECTION, AND RESTART THE GAME.'/1,' SELECT AND
292 1/,' ENTER YOUR PASSWORD')
293 READ(5,200) INPUT
294 CALL INVRT61(INPUT,BLUPAS)
295 WRITE(6,123) BLUPAS
296 123 FORMAT(1,' YOUR PASSWORD WILL BE ','BLU.'-, IF YOU ARE',
297 1/,' READY TO CONTINUE TYPE Y(ES). TYPE N(0) IF YOU WISH TO',
298 1/,' CHANGE YOUR PASSWORD.')
299 READ(5,200) INPUT
300 CALL INVRT1(INPUT,A)
IF A.EQ.ALPHA(2) GO TO 5
WRITE(6,124)
FORMAT///,'ENTER THE NUMBER OF BLUE SURFACE SHIPS.'
READ(5,200) INPUT
CALL NIOVRAT(INPUT,BLUSUR)
WRITE(6,126)
FORMAT///,'ENTER THE NUMBER OF BLUE SUBMARINES.'
READ(5,200) INPUT
CALL NONVRAT(INPUT,BLUSUB)
WRITE(6,127)
FORMAT///,'ENTER THE NUMBER OF BLUE SHIPS THAT MUST',
1/.' BE DESTROYED FOR PREMATURE GAME ENDING.'
READ(5,200) INPUT
CALL NONVRAT(INPUT,BLUDES)
WRITE(6,128) BLUSUR,BLUSUB,BLUDES
FORMAT///,'ECO'.///,'BLUE SURFACE SHIPS - ',14,
1/.' BLUE SUBMARINES - ',14,
1/.' NUMBER OF BLUE SHIPS DESTROYED TO END GAME - ',14
WRITE(6,129)
FORMAT///,'TYPE Y(ES) IF THESE VALUES ARE CORRECT. OR N(O) IF ',
1/.' YOU WANT TO REENTER THEM.'
READ(5,200) INPUT
CALL INVAT1(INPUT,A)
IF A.EQ.ALPHA(2) GO TO 5
WRITE(6,130)
FORMAT///'THE FOLLOWING INPUTS APPLY ONLY TO THE RED PLAYER'.
1/.' THE FIRST IS THE RED PLAYERS PASSWORD WHICH IS A ',
1/.' MAXIMUM OF 6 CHARACTERS. THE PASSWORD IS IMPORTANT BECAUSE',
1/.' IT IS REQUIRED PRIOR TO THE INPUT OF COMBAT, MOVE AND SEARCH'.
1/.' COMMANDS. IF YOU FORGET THE PASSWORD, YOU WILL HAVE TO RETUR
IN'.///' TO MENU SELECTION, AND RESTART THE GAME.'///' SELECT AND
1/.' ENTER YOUR PASSWORD.'
READ(5,200) INPUT
CALL INVRAT6(INPUT,REDPAS)
WRITE(6,131) REDPAS
FORMAT///,'YOUR PASSWORD WILL BE - ',6A1,'- IF YOU ARE',
1/.' READY TO CONTINUE TYPE Y(ES). TYPE N(O) IF YOU WISH TO',
1/.' CHANGE YOUR PASSWORD.'
READ(5,200) INPUT
CALL INVAT1(INPUT,A)
IF A.EQ.ALPHA(2) GO TO 6
WRITE(6,132)
FORMAT///,'ENTER THE NUMBER OF RED SURFACE SHIPS.'
READ(5,200) INPUT
CALL NIOVRAT(INPUT,REDSUR)
WRITE(6,135)
FORMAT///,'ENTER THE NUMBER OF RED SUBMARINES.'
READ(5,200) INPUT
CALL NONVRAT(INPUT,REDSUB)
WRITE(6,136)
FORMAT(1X,'ENTER THE NUMBER OF RED SHIPS THAT MUST':
1X,' BE DESTROYED FOR PREMATURE GAME ENDING.')
READ(5,200) INPUT
CALL INVATI(INPUT,REDDES)
WRITE(6,137) REDSUR,REDSUB,REDDES

FORMAT(1X,' ECHO .1X,' RED SURFACE SHIPS - ',14.
1X,' RED SUBMARINES - ',14.
1X,' NUMBER OF RED SHIPS DESTROYED TO END GAME - ',14)
WRITE(6,138)

FORMAT(1X,'DOTYPE YES IF THESE VALUES ARE CORRECT, OR NO IF',
1X,' YOU WANT TO REENTER THEM.')
READ(5,200) INPUT
CALL INVATI(INPUT,A)
IF(A.EQ.0) GO TO 6

C
SET VARIABLES TO BE USED AS POINTERS TO THE PLATFORM ARRAY
C
PASUZ=MAXPLT
PASUBI=MAXPLT-REDSUB+1
PBSUB2=PBSUB1-1
PBSUB2=PBSUB2-BLUSUB+1
PASUB2=PASUB1-1
PASUB1=PASUB2-REDSUR+1
PBSUR2=PBSUR1-1
PBSUR1=PBSUR2-BLUSUR+1

C
LOAD BLUE WEAPON SYSTEMS
C
WRITE(6,140)
FORMAT(1X,' YOU ARE NOW READY TO LOAD THE BLUE WEAPON SYSTEMS.',
1X,' ENTER THE BLUE PASSWORD.')
READ(5,200) INPUT
CALL INVATI(INPUT,CHECK)
A=0
GO 141 I=1,6
IF(CHECK(I).NE.BLUPAS(I)) A=1
IF(A.EQ.1) GO TO 142
CONTINUE
IF(A.NE.1) GO TO 8
WRITE(6,143)
FORMAT(1X,' YOU HAVE MADE AN ERROR IN YOUR PASSWORD. TYPE YES',
1X,' IF YOU WANT TO TRY AGAIN. OTHERWISE YOU WILL BE',
1X,' RETURNED TO MENU SELECTION.')
READ(5,200) INPUT
CALL INVATI(INPUT,A)
IF(A.EQ.0) GO TO 7
GO TO 1
IF(BLUSUR.EQ.0) GO TO 170
DO 165 SHIP=PBSUR1,PBSUR2
SHIPNO=SHIPNO+1
WRITE(6,144) SHIPNO
FORMAT('BLUE SURFACE SHIP NUMBER ',',/,' ENTER A 6 CHARACTER',
1' NAME FOR THE SHIP.')
READ(5,200) INPUT
CALL INVRT6UINPUT,CHECK
DO 145 J=1,6
CALL SHIPNO+J
WRITE(6,144) CHECK(J)
CONTINUE
WRITE(6,146) CHECK
FORMAT('ENTER THE SHIP CLASS OF ',',/.'
READ(5,202) CLASNA
202 FORMAT(8A2)
CALL SCLASS(CLASNA,CLASI,NAME)
IF(CLASI.EQ.0) WRITE(6,148)
FORMAT('YOU HAVE MADE A MISTAKE. ENTER THE SHIP CLASS WITH NO',
1/,' LEADING BLANKS USING A PROPER CLASS NAME. ')
IF(CLASI.EQ.0) GO TO 147
PLATFM(SHIP,3)=CLASI
WRITE(6,149) CHECK
FORMAT('ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL ',
READ(5,200) INPUT
CALL NONVRT(INPUT,A)
PLATFM(SHIP,3)=A
PLATFM(SHIP,8)=1
PLATFM(SHIP,20)=SHIPDA(CLASI,18)
PLATFM(SHIP,12)=SHIPDA(CLASI,11)
PLATFM(SHIP,13)=SHIPDA(CLASI,2)
PLATFM(SHIP,14)=SHIPDA(CLASI,3)
PLATFM(SHIP,15)=SHIPDA(CLASI,9)
PLATFM(SHIP,16)=SHIPDA(CLASI,11)
PLATFM(SHIP,17)=SHIPDA(CLASI,13)
PLATFM(SHIP,18)=SHIPDA(CLASI,6)
PLATFM(SHIP,19)=SHIPDA(CLASI,7)
PLATFM(SHIP,21)=SHIPDA(CLASI,8)
PLATFM(SHIP,22)=SHIPDA(CLASI,10)
PLATFM(SHIP,23)=SHIPDA(CLASI,12)
PLATFM(SHIP,24)=SHIPDA(CLASI,1)
PLATFM(SHIP,25)=SHIPDA(CLASI,5)
PLATFM(SHIP,26)=SHIPDA(CLASI,14)
PLATFM(SHIP,27)=SHIPDA(CLASI,16)
PLATFM(SHIP,28)=SHIPDA(CLASI,15)
PLATFM(SHIP,29)=SHIPDA(CLASI,17)
C
C
LOAD BLUE WEAPON SYSTEMS - AIRCRAFT
C
DO 155 I=26,27
IF (PLATFM(SHIP,1).EQ.0) GO TO 165
ACINDX=PLATFM(SHIP,1)
WRITE(6,151) CLASSNA, PLATFM(SHIP,1+2), (NAMEAC(ACINDX,J),J=1,6)
FORMAT(1//,'THE ',6A2,' CLASS OF SHIP HAS ',12,IX,6A2,IX,' TYPE',')
PLATFM(SHIP,1+5)=BLURIR+1
BLUIR=BLUIR+PLATFM(SHIP,1+2)
PLATFM(SHIP,1+7)=BLUIR
ACOUNT=0
P=PLATFM(SHIP,1+5)
Q=PLATFM(SHIP,1+7)
DO 154 AC=P,Q
ACOUNT=ACOUNT+1
WRITE(6,152) ACOUNT
FORMAT(1//,'ENTER A 6 CHARACTER NAME FOR AIRCRAFT ',12)
READ(15,'(O0)') INPUT
CALL INVT61(INPUT,CHECK)
DO 153 I=1,6
NAME(AC,J)=CHECK(J)
CONTINUE
PLATFM(AC,3)=PLATFM(SHIP,3)
PLATFM(AC,8)=0
PLATFM(AC,12)=ACDATA(ACINDX,1)
PLATFM(AC,13)=ACDATA(ACINDX,2)
PLATFM(AC,14)=ACDATA(ACINDX,3)
PLATFM(AC,15)=ACDATA(ACINDX,5)
PLATFM(AC,16)=ACDATA(ACINDX,7)
PLATFM(AC,17)=ACDATA(ACINDX,4)
PLATFM(AC,18)=ACDATA(ACINDX,6)
PLATFM(AC,20)=1
PLATFM(AC,21)=SHIP
PLATFM(AC,22)=PLATFM(SHIP,11)
CONTINUE
CONTINUE
CONTINUE
CONTINUE
C
C LOAD BLUE WEAPON SYSTEMS - SUBMARINES
C
IF (BLUSUB.EQ.0) GO TO 10
SUBNO=0
DO 175 SUB=PBSUB1,PBSUB2
SUBNO=SUBLNO+1
WRITE(6,171) SUBNO
FORMAT(1//,'BLUE SUBMARINE NUMBER ',14,' ENTER A 6 DIGIT',')
I ' CHARACTER NAME FOR THE SUBMARINE.')
READ(15,200) INPUT
CALL INVT61(INPUT,CHECK)
DO 172 J=1,6
NAME(SUB,J)=CHECK(J)
CONTINUE
501 174 WRITE (6, 173) CHECK
502 173 FORMAT ('ENTER THE SUBMARINE CLASS OF ', BA1)
503 READ (5, 202) CLASNA
504 CALL SCLASS (CLASNA, CLASIN, NAMESH)
505 IF (CLASIN.EQ.0) WRITE (6, 148)
506 IF (CLASIN.EQ.0) GO TO 174
507 PLATFM (SUB, 30) = CLASIN
508 WRITE (6, 176) CHECK
509 176 FORMAT ('ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL ',
510 LOCATION OF ', BA1)
511 READ (5, 200) INPUT
512 CALL NONVRT (INPUT, A)
513 PLATFM (SUB, 3) = A
514 PLATFM (SUB, 8) = 2
515 PLATFM (SUB, 20) = SHIPDA (CLASIN, 18)
516 PLATFM (SUB, 12) = SHIPDA (CLASIN, 1)
517 PLATFM (SUB, 13) = SHIPDA (CLASIN, 2)
518 PLATFM (SUB, 14) = SHIPDA (CLASIN, 3)
519 PLATFM (SUB, 15) = SHIPDA (CLASIN, 9)
520 PLATFM (SUB, 16) = SHIPDA (CLASIN, 11)
521 PLATFM (SUB, 17) = SHIPDA (CLASIN, 13)
522 PLATFM (SUB, 18) = SHIPDA (CLASIN, 6)
523 PLATFM (SUB, 19) = SHIPDA (CLASIN, 7)
524 PLATFM (SUB, 21) = SHIPDA (CLASIN, 8)
525 PLATFM (SUB, 22) = SHIPDA (CLASIN, 10)
526 PLATFM (SUB, 23) = SHIPDA (CLASIN, 12)
527 PLATFM (SUB, 24) = SHIPDA (CLASIN, 4)
528 PLATFM (SUB, 25) = SHIPDA (CLASIN, 5)
529 PLATFM (SUB, 26) = SHIPDA (CLASIN, 14)
530 PLATFM (SUB, 27) = SHIPDA (CLASIN, 16)
531 PLATFM (SUB, 28) = SHIPDA (CLASIN, 15)
532 PLATFM (SUB, 29) = SHIPDA (CLASIN, 17)
533 175 CONTINUE
534 C
535 C LOAD RED WEAPON SYSTEMS
536 C
537 C WRITE (6, 240)
538 240 FORMAT ('YOU ARE NOW READY TO LOAD THE RED WEAPON SYSTEMS. '
539 /, ' ENTER THE RED PASSWORD. ')
540 241 READ (5, 200) INPUT
541 CALL INVRTB (INPUT, CHECK)
542 A = 0
543 DO 241 I = 1, 6
544 IF (CHECK (I) .NE. REDPAS (I)) A = 1
545 IF (A .EQ. 1) GO TO 242
546 CONTINUE
547 242 IF (A .NE. 1) GO TO 11
548 243 WRITE (6, 243)
549 243 FORMAT ('YOU HAVE MADE AN ERROR IN YOUR PASSWORD. TYPE Y(ES). '
197
551 IF YOU WANT TO TRY AGAIN. OTHERWISE YOU WILL BE
552 RETURNED TO MENU SELECTION."
553 READ(5,200) INPUT
554 CALL INVRTI(INPUT,A)
555 IF(A.EQ.ALPHA(1)) GO TO 10
556 GO TO 1
557 11 REDRIA=BLUAR
558 IF(REDSRA.EQ.0) GO TO 270
559 SHIPND=0
560 DO 265 SHIP=FRSAU1,FRSUA2
561 SHIPND=SHIPND+1
562 WRITE(6,244) SHIPND
563 244 FORMAT('REO SURFACE SHIP NUMBER ',14,' ENTER A 6 CHARACTER',
564 1. NAME FOR THE SHIP.')
565 READ(5,200) INPUT
566 CALL INVRT6(INPUT,CHECK)
567 DO 245 J=1,6
568 NAME(SHIP,J)=CHECK(J)
569 245 CONTINUE
570 246 WRITE(6,246) CHECK
571 246 FORMAT(*ENTER THE SHIP CLASS OF ',6AI)
572 READ(5,203) CLASNA
573 203 FORMAT(8A2)
574 CALL SCLASS(CLASNA,CLASIN,NAMESH)
575 IF(CLASIN.EQ.0) WRITE(6,248)
576 248 FORMAT('YOU HAVE MADE A MISTAKE. ENTER THE SHIP CLASS WITH NO',
577 ' LEADING BLANKS USING A PROPER CLASS NAME.')
578 IF(CLASIN.EQ.0) GO TO 247
579 PLATFM(SHIP,30I-CLASIN
580 WRITE(6,249) CHECK
581 249 FORMAT('ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL ',
582 1/, LOCATION OF ',6AI)
583 READ(5,200) INPUT
584 CALL INVRT(INPUT,A)
585 PLATFM(SHIP,3)=A
586 586 PLATFM(SHIP,8)=1
587 PLATFM(SHIP,2D)=SHIPDA(CLASIN,18)
588 PLATFM(SHIP,12)=SHIPDA(CLASIN,11)
589 PLATFM(SHIP,13)=SHIPDA(CLASIN,2)
590 PLATFM(SHIP,14)=SHIPDA(CLASIN,3)
591 PLATFM(SHIP,15)=SHIPDA(CLASIN,9)
592 PLATFM(SHIP,16)=SHIPDA(CLASIN,11)
593 PLATFM(SHIP,17)=SHIPDA(CLASIN,13)
594 PLATFM(SHIP,18)=SHIPDA(CLASIN,6)
595 PLATFM(SHIP,19)=SHIPDA(CLASIN,7)
596 PLATFM(SHIP,21)=SHIPDA(CLASIN,8)
597 PLATFM(SHIP,22)=SHIPDA(CLASIN,10)
598 PLATFM(SHIP,23)=SHIPDA(CLASIN,12)
599 PLATFM(SHIP,24)=SHIPDA(CLASIN,4)
600 PLATFM(SHIP,25)=SHIPDA(CLASIN,5)
601 PLATFM(SHIP,26)=SHIPOA(CLASIN,14)
602 PLATFM(SHIP,27)=SHIPOA(CLASIN,16)
603 PLATFM(SHIP,28)=SHIPOA(CLASIN,15)
604 PLATFM(SHIP,29)=SHIPOA(CLASIN,17)
605 C
606 C
607 C LOAD RED WEAPON SYSTEMS - AIRCRAFT
608 C
609 250 DO 255 I=26,27
610 IF (PLATFM(SHIP, I).EQ.0) GO TO 265
611 ACINDX=PLATFM(SHIP, I)
612 WRITE(6,251) CLASIN, PLATFM(SHIP, I+2), NAMEAC(ACINDX, J), J=1,8
613 251 FORMAT(//,'OTHER "BA2"," CLASS OF SHIP HAS ',12,IX,6A2, IX,' TYPE ',
   1 ' AIRCRAFT.' )
614 PLATFM(SHIP, I+5)=REDAIR*1
615 REDAIR=REDAIR*PLATFM(SHIP, I+2)
616 PLATFM(SHIP, I+7)=REDAIR
617 ACOUNT=0
618 P=PLATFM(SHIP, I+5)
619 Q=PLATFM(SHIP, I+7)
620 DO 254 AC=P,Q
621 DO 254 ACOUNT=ACOUNT+1
622 WRITE(6,252) ACOUNT
623 252 FORMAT(//,'ENTER A 6 CHARACTER NAME FOR AIRCRAFT ',12)
624 READ(5,200) INPUT
625 CALL INVRT6(INPUT,CHECK)
626 DO 253 J=1,8
627 NAME(AC,J)=CHECK(J)
628 253 CONTINUE
629 250 CONTINUE
630 PLATFM(AC,3)=PLATFM(SHIP, 3)
631 PLATFM(AC,8)=0
632 PLATFM(AC,12)=ACDATA(ACINDX, 11)
633 PLATFM(AC,13)=ACDATA(ACINDX, 12)
634 PLATFM(AC,14)=ACDATA(ACINDX, 13)
635 PLATFM(AC,15)=ACDATA(ACINDX, 15)
636 PLATFM(AC,16)=ACDATA(ACINDX, 16)
637 PLATFM(AC,17)=ACDATA(ACINDX, 17)
638 PLATFM(AC,18)=ACDATA(ACINDX, 18)
639 PLATFM(AC,20)=1
640 PLATFM(AC,21)=SHIP
641 PLATFM(AC,22)=PLATFM(SHIP, 1)
642 254 CONTINUE
643 255 CONTINUE
644 265 CONTINUE
645 C
646 C LOAD RED WEAPON SYSTEMS - SUBMARINES
647 C
648 270 IF (REDSUB.EQ.0) GO TO 15
649 SUBNO=0 DO 275 SUB=SUB1,SUB2
651 SUBNO = SUBNO + 1
652 WRITE (6, 271) SUBNO
653 271 FORMAT (1X, 'READ SUBMARINE NUMBER ', A14, ' ENTER A 6 DIGIT',
654 ' CHARACTER NAME FOR THE SUBMARINE.')
655 READ (5, 200) INPUT
656 CALL INVRTS (INPUT, CHECK)
657 DO 272 J = 1, 6
658 NAME (SUB, J) = CHECK (J)
659 272 CONTINUE
660 WRITE (6, 273) CHECK
661 273 FORMAT (1X, 'ENTER THE SUBMARINE CLASS OF ', A6)
662 READ (5, 203) CLASNA
663 CALL SCLASS (CLASNA, CLASIN, NAMESH)
664 IF (CLASIN.EQ.0) WRITE (6, 248)
665 IF (CLASIN.EQ.0) GO TO 274
666 PLATFM (SUB, 30) = CLASIN
667 WRITE (6, 276) CHECK
668 276 FORMAT (1X, 'ENTER THE 4 DIGIT HEX NUMBER OF THE INITIAL ',
669 '1/., LOCATION OF ', A6)
670 READ (5, 200) INPUT
671 CALL NONVRT (INPUT, A)
672 PLATFM (SUB, 3) = A
673 PLATFM (SUB, 8) = 2
674 PLATFM (SUB, 20) = SHIPOA (CLASIN, 18)
675 PLATFM (SUB, 12) = SHIPOA (CLASIN, 1)
676 PLATFM (SUB, 13) = SHIPOA (CLASIN, 2)
677 PLATFM (SUB, 14) = SHIPOA (CLASIN, 3)
678 PLATFM (SUB, 15) = SHIPOA (CLASIN, 9)
679 PLATFM (SUB, 16) = SHIPOA (CLASIN, 16)
680 PLATFM (SUB, 17) = SHIPOA (CLASIN, 13)
681 PLATFM (SUB, 18) = SHIPOA (CLASIN, 6)
682 PLATFM (SUB, 19) = SHIPOA (CLASIN, 7)
683 PLATFM (SUB, 21) = SHIPOA (CLASIN, 8)
684 PLATFM (SUB, 22) = SHIPOA (CLASIN, 10)
685 PLATFM (SUB, 23) = SHIPOA (CLASIN, 12)
686 PLATFM (SUB, 24) = SHIPOA (CLASIN, 4)
687 PLATFM (SUB, 25) = SHIPOA (CLASIN, 5)
688 PLATFM (SUB, 26) = SHIPOA (CLASIN, 14)
689 PLATFM (SUB, 27) = SHIPOA (CLASIN, 16)
690 PLATFM (SUB, 28) = SHIPOA (CLASIN, 15)
691 PLATFM (SUB, 29) = SHIPOA (CLASIN, 17)
692 275 CONTINUE
693 C SET VARIABLES TO BE USED AS POINTERS TO THE PLATFORM ARRAY
694 C
695 15 PBAIR1 = 1
696 PB1R2 = BLUAIR
697 READ = REDAIR = BLUAIR
698 PRAIR1 = BLUAIR + 1
699 PRAIR2 = PRAIR1 + REDAIR - 1
IAIR=BLUAIR+REDAIR
ISURF=BLUSUR+REDSUR
ISUB=BLUSUB+REDSUB
IPLAT=IAIR=ISURF+ISUB
IF(IPLAT.LE.MAXPLT) GO TO 16
WRITE(6,280)
280 FORMAT('THE TOTAL NUMBER OF WEAPON SYSTEMS HAS EXCEEDED THE MAXIMUM NUMBER OF PLATFORMS ALLOWED. YOU WILL HAVE TO RESTART THE GAME.')
GO TO 99
C C LOOP TO CLOSE PLATFM AND REASSIGN POINTERS
C
15 IF(IPBSUR1.EQ.(PRAIR2+1)) GO TO 17
ISHIP=ISURF+ISUB
DO 283 I=I,ISHIP
DO 281 J=1,34
PLATFM(PRAIR2+1,J)=PLATFM(IPBSUR1+1,J)
CONTINUE
281 DO 282 J=1,6
NAME(PRAIR2+1,J)=NAME(IPBSUR1+1,J)
CONTINUE
282 DO 284 J=28,29
IF(PLATFM(PRAIR2+1,J).EQ.0) GO TO 283
P=PLATFM(PRAIR2+1,J+3)
Q=PLATFM(PRAIR2+1,J+5)
DO 285 K=P.O
CONTINUE
285 CONTINUE
284 CONTINUE
283 CONTINUE
DO 286 I=1,1PLAT
BUFFER(I,4)=PLATFM(I,3)
CONTINUE
286 CONTINUE
PBSUR1=PRAIR2*1
PBSUR2=PBSUR1+BLUSUR-1
PRSUR1=PBSUR2*1
PRSUR2=PRSUR1+REDSUR-1
PBSUB1=PRSUR2*1
PBSUB2=PBSUB1+BLUSUB-1
PRSUB1=PBSUB2*1
PRSUB2=PRSUB1+REDSUB-1
C C CONTROL OF GAME TURNS
C
17 ITURN=0
GO TO 19
19 WRITE(6,300)
300 FORMAT('BLUE CALL TO STATUS ROUTINES. ENTER THE BLUE PASSWORD.')
READ(5,200) INPUT
CALL INVRT6(INPUT,CHECK)
CALL STATUSICHECK,ERROR,NAMWP)

IF ERROR.EQ.0) GO TO 20

WRITE(6,301)

IF ERROR.EQ.0) GO TO 21

WRITE(6,302)

READ(5,200) INPUT

CALL INVRTI(INPUT, A)

IF (A.EQ.ALPHA(21) GO TO 1

GO TO 19

WRITE(6,303)

READ(5,200) INPUT

CALL INVRTI(INPUT, CHECK)

CALL STATUSICHECK,ERROR,NAMWP)

IF ERROR.EQ.0) GO TO 21

WRITE(6,304)

READ(5,200) INPUT

CALL INVRTI(INPUT, A)

IF (A.EQ.ALPHA(21) GO TO 1

GO TO 20

CALL CTRL

BDES=0

RDES=D

DO 303 I=1,100

S=SUNK(I)

IF (S.EQ.-1) GO TO 22

IF (S.GE.PBSURI.AND.S.LE.PBSUR2) .OR. (S.GE.PSUB1.AND.S.LE.PSUB2)

IBDES=IBDES+1

IF (S.GE.PRSURI.AND.S.LE.PRSUR2).OR. (S.GE.PRSUB1.AND.S.LE.PRSUB2)

IBDES=IBDES+1

CONTINUE

IF (IBDES.EQ.BLUSUR*BLUSUB1) GO TO 9997

IF (IBDES.EQ.REDDES*REDSUB1) GO TO 9997

IF (IBDES.EQ.REDDES+1) GO TO 23

WRITE(6,305)

FORMAT('THE NUMBER OF BLUE SHIPS DESTROYED IS EQUAL TO OR',

'GREATER THAN THE NUMBER INPUT FOR GAME END. IF YOU WANT',

'TO CONTINUE THE GAME TYPE YES. TYPE NO TO END THE GAME',

READ(5,200) INPUT

CALL INVRTI(INPUT, A)

IF (A.EQ.ALPHA(21) GO TO 9997

IF (A.EQ.ALPHA(21) GO TO 9997

IF (A.EQ.ALPHA(21) GO TO 9997

WRITE(6,306)

FORMAT('THE NUMBER OF RED SHIPS DESTROYED IS EQUAL TO OR',

'GREATER THAN THE NUMBER INPUT FOR GAME END. IF YOU WANT',

'TO CONTINUE THE GAME TYPE YES. TYPE NO TO END THE GAME',

READ(5,200) INPUT

CALL INVRTI(INPUT, A)
IF (A.EQ. ALPH A(2))  GO TO 997
   TURN=TURN+1
203 IF (TURN.LT.TURN)  GO TO 25
WRITE (6,304) TURN
304 FORMAT ('YOU HAVE JUST COMPLETED GAME TURN ','.',TURN,'.'.)
      TYPE YES! IF YOU WANT TO CONTINUE THE GAME, OR NO!'
      TO ALLOW THE GAME TO END.')
HEAD(5,200) INPUT
604 CALL INPUT (INPUT, A)
GO TO 997
25
GO TO 1
C
GAME END
C
WRITE (6,307)
307 FORMAT ('BLUE RESULTS:','//')
WRITE (6,308)
308 FORMAT ('ON ',3X,'FINAL','/',5X,'SHIP','/5X,'STATUS/HITS','/5X,'LOCATION:')
COUNT=0
IF (PLUSUB.EQ.0)  GO TO 27
A=PLUSUB
B=PLUSUB
C=1
26
GO 313 PLT=A,B
TEMP=PLATFM(PLT,11)
27 IF (PLATFM(PLT,20).EQ.0)  WRITE (6,309) (NAME(PLT,J),J=1,61)
309 FORMAT (A,15X)
ITEM=PLATFM(PLT,3)
310 FORMAT ('ON ',3X,'FINAL','/',5X,'SHIP','/5X,'STATUS/HITS','/5X,'LOCATION:')
COUNT=COUNT+1
IF (COUNT.EQ.0)  GO TO 313
311 WRITE (6,311)
311 FORMAT ('ONE MORE ON NEXT PAGE. TYPE YES TO CONTINUE. ')
READ(5,2011) G
WRITE (6,312)
FORMAT (3X,'FINAL','/',5X,'SHIP','/5X,'STATUS/HITS','/5X,'LOCATION:')
CONTINUE
403 IF (C.GT.3)  GO TO 998
404 IF (C.GT.2)  GO TO 29
405 IF (C.GT.1)  GO TO 29
27 IF (PLUSUB.EQ.0)  GO TO 28
407 A=PLUSUB
408 B=PLUSUB
409 C=2
410 GO TO 26
203
READ 5,201 G
WRITE 6,315
FORMAT ('RED RESULTS',//)
WRITE 6,100
COUNT=0
IF RED=1 .EQ. 0 GO TO 29
A=PASUR1
B=PASUR2
C=3
GO TO 26
IF REDSUB.EQ.0) GO TO 9998
A=PASUB1
B=PASUB2
C=4
GO TO 26
9998 WRITE (6,317)
FORMAT ('EXITING THE GAME.'
READ 5,200 INPUT
CALL INRTI(INPUT,A)
IF AI.EQ.ALAPHA(1)) GO TO 1
9999 WRITE (6,318)
FORMAT ('EXITING THE GAME.'
STOP
END
SUBROUTINE STATUS(PASWD, ERROR, NAMEWP)

C THIS SUBROUTINE IS THE DRIVER FOR ALL DISPLAY ROUTINES.

C IMPLICIT INTEGER (A-Z)
DIMENSION PASWD(6), INPUT(70), ALPHA(2), NAMEWP(29, 6)
COMMON PLATFORM(100, 37), DETECT(100, 41), NEF(29, 31), HIT(100, 31), SUNK(100)
1, WATER(100, 61), BUFFER(100, 81), BLUPAS(61), REDPAS(61),
ZPLAT, ASUR1, ISUB, ADV, REV, ERATIC, BLUAIR, REDAIR, BLUSUR,
3REDUR, 4USUB, REDSUB, PAIR1, PAIR2, PAIR1, PAIR2, PBSUR1, PBSUR2,
WPASUR1, PASUR2, PBSUB1, PBSUB2, PASUB1, PASUB2
DATA APLA/"T", NA/0
R-0
ERROR-O
DO 100 I=1, 6
IF(PASWD(I).NE.BLUPAS(I)) A=1
IF(A.EQ.1) GO TO 200
100 CONTINUE
200 IF(W.EQ.1) GO TO 300
NC1=1
AC2=BLUAIR
SUR1=PBSUR1
SUR2=PBSUR2
SUB1=PBSUB1
SUB2=PBSUB2
AC=BLUAIR
SUR1=BLUSUR
SUR2=BLUSUB
SIDE=1
300 GO TO 800
31300
AC=0
DO 400 I=1, 6
IF(PASWD(I).NE.REDPAS(I)) A=1
IF(A.EQ.1) GO TO 500
400 CONTINUE
500 IF(W.EQ.1) GO TO 600
AC1=PAIR1
AC2=PAIR2
SUR1=PRSUR1
SUR2=PRSUR2
SUB1=PRSUB1
SUB2=PRSUB2
AC=REDAIR
SUR1=REDUSR
SUR1=REDUSUB
SIDE=2
600 ERROR=1
600 ERROR=1
WRITE (6, 700)
700 FORMAT(1DYOU HAVE MADE AN ERROR IN THE ENTRY OF YOUR PASSWORD.)

205
GO TO 9999
800     WRITE(6,8211)
821  FORMAT('THE FOLLOWING DISPLAY OPTIONS ARE AVAILABLE TO YOU:',
5///. ' 1. DETECTIONS OF ENEMY PLATFORMS.',
5///. ' 2. RESULTS OF WEAPONS FIRED.',
5///. ' 3. DAMAGE TO FRIENDLY PLATFORMS.',
5///. ' 4. STATUS OF FRIENDLY PLATFORMS AND INPUT OF INSTRUCTIONS.',
5///. ' 5. BEGIN NEXT GAME TURN.')
1000  WRITE(6,824)
824  FORMAT('SELECT AND ENTER THE NUMBER OF A DISPLAY OPTION.
1///. ENTER A 6 FOR A LIST OF DISPLAY OPTIONS.')
     READ 5,8221 INPUT
822  FORMAT('70A11')
     CHILL INVAT1(INPUT,A1)
     IF(A.EQ.61 GO TO 800
     IF(A.EQ.1.or.A.EQ.2.or.A.EQ.3.or.A.EQ.4.or.A.EQ.51 GO TO 830
     WRITE(6,823)
823  FORMAT('YOU HAVE MADE AN ERROR IN YOUR INPUT SELECTION. TRY AGAIN IN.')
     GO TO 1000
830  IF(A.GT.1) GO TO 2
     CHILL DETXT(AC1,AC2,SUR1,SUR2,SUB1,SUB2,NAMWP)
     GO TO 1000
2  IF(A.GT.2) GO TO 3
     CHILL SHOTRE(AC1,AC2,SUR1,SUR2,SUB1,SUB2,NAMWP,AC,SUR,SUB)
     GO TO 1000
3  IF(A.GT.3) GO TO 4
     CHILL DAMRE(SUR1,SUR2,SUB1,SUB2,SUR,SUB)
     GO TO 1000
4  IF(A.GT.4) GO TO 5
     CHILL STATIN(AC1,AC2,SUR1,SUR2,SUB1,SUB2,NAMWP)
     GO TO 1000
5  WRITE(6,831)
831  FORMAT('THIS OPTION WILL BEGIN THE NEXT GAME TURN. TYPE YES.')
     IF(Y) IF YOU HAVE ENTERED ALL YOUR COMBAT, MOVE AND SEARCH .
     IF(Y) INSTRUCTIONS AND YOU ARE READY TO COMMENCE PLAY. TYPE NO.
     IF(Y) TO RETURN TO DISPLAY SELECTION.
     R.ADI(5,932) INPUT
     CHILL INVAT1(INPUT,B1)
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
     RETURN
    phanumeric
SUBROUTINE DETXTHACI.AC2.SUIt.SUII.SU2.NANEWP

C THIS SUBROUTINE DISPLAYS THE DETECTIONS OF THE CALLING FORCE.

C

IMPLICIT INTEGER (A-Z)

COMMON PLATFORM(100,37), DETECT(100,4), WEP(29,3), HIT(100,3), SUNK(100)

1, WATER(100,6), NAME(100,6), BUFFER(100,8), BLUPAS(6), REDPAS(6),
2 PLANF, IRR, ISURF, ISUB, ADV, PLO, RE, PEV, ERATIC, BLUARI, REDAIR, BLUSUR,
3 REDSUN, BLUSUB, REDSUB, PBRAIN, PBAIR2, PBRAIN, PRAI2, PBSPUR, PBSPUR2,
4 PBSPU(3), PBSPUR2, PBSPUR1, PBSPUR2, PBSPUR, PBSPU2

DIMENSION ALPHA1(2,5), ALPHA2(2,4), NAMEWP(29,6)

DATA ALPHA1/'SU', 'SU', 'AR', 'AR', 'IN', 'E', 'E'/

DATA ALPHA2/'PA', 'PA', 'SS', 'TI', 'TV', 'VE', 'E', 'E'/

COUNT=0

WRITE(6,1)

FORMAT('I',14X,'SONAR',7X,'SHIP',6X,'DETECTING',6X,'TYPE',8X,
1 'TYPE',6X,'PLATFORM',5X,'USED',6X,'DETECTED',3X,'LOCATION',/)

G0 5 PLT=1,100

D1=DETECT(PLT,1)

D2=DETECT(PLT,2)

IF(D1.EQ.1) GO TO 9999

IF(D1.GE.AC1.AND.D1.LE.AC21.AND.D2.LE.SUR2) GO TO 2

GO TO 6

WRITE(6,3) (NAME(DI,J),J=1,6), (ALPHA1(DETECT(PLT,3),I),I=1,4),

1 (ALPHA1(PLATF,4),J),J=1,5)

FORMAT('I',3X,5A1,5X,4A2,3X,5A2,3X,14)

COUNT=COUNT+1

IF(COUNT.LT.19) GO TO 6

COUNT=0

WRITE(6,4)

FORMAT('MORE DETECTIONS ON NEXT PAGE. TYPE Y(ES) TO CONTINUE.')

READ(5,5) C

WRITE(6,11)

FORMAT('I',14X,'-platform',14X,'TYPE',8X,
1 'platform',6X,'used',6X,'detected',3X,'location',/)

G0 5 PLT=2,100

D1=DETECT(PLT,1)

D2=DETECT(PLT,2)

IF(D1.EQ.1) GO TO 9999

IF(D1.GE.AC1.AND.D1.LE.AC21.AND.D2.LE.SUR2) GO TO 2

GO TO 6

WRITE(6,3) (NAME(DI,J),J=1,6), (ALPHA2(DETECT(PLT,3),I),I=1,4),

1 (ALPHA2(PLATF,4),J),J=1,5)

FORMAT('I',3X,5A1,5X,4A2,3X,5A2,3X,14)

COUNT=COUNT+1

IF(COUNT.LT.19) GO TO 6

COUNT=0

WRITE(6,4)

FORMAT('END OF DETECTION ROUTINE. TYPE Y(ES) TO CONTINUE.')

READ(5,5) C

CONTINUE

9999 WRITE(6,7)

FORMAT('I',14X,'platform',14X,'TYPE',8X,
1 'platform',6X,'used',6X,'detected',3X,'location',/)

G0 5 PLT=3,100

D1=DETECT(PLT,1)

D2=DETECT(PLT,2)

IF(D1.EQ.1) GO TO 9999

IF(D1.GE.AC1.AND.D1.LE.AC21.AND.D2.LE.SUR2) GO TO 2

GO TO 6

WRITE(6,3) (NAME(DI,J),J=1,6), (ALPHA3(DETECT(PLT,3),I),I=1,4),

1 (ALPHA3(PLATF,4),J),J=1,5)

FORMAT('I',3X,5A1,5X,4A2,3X,5A2,3X,14)

COUNT=COUNT+1

IF(COUNT.LT.19) GO TO 6

COUNT=0

WRITE(6,4)

FORMAT('END OF DETECTION ROUTINE. TYPE Y(ES) TO CONTINUE.')

READ(5,5) C

RETURN

END
SUBROUTINE SHOTRELACIAC2, SUR1, SUR2, SUB1, SUB2, NAMEWP, AC, SUR, SUBI
C
THIS SUBROUTINE DISPLAYS THE RESULTS OF WEAPONS FIRED BY THE
CALLING FORCE.

IMPLICIT INTEGER (A-Z)
REAL R
COMMON PLATFM(100, 37), DETECT(100, 41), WEP(29, 31), HIT(100, 31), SUNK(100)
I, WATER(100), NAME(100, 61), BUFFER(100, 61), BLUPAS(61), REDPAS(61),
2IPLAT, IRAI, ISURF, ISUB, ADV, POV, PEV, ERATIC, BLAIR, REDAIR, BLUSUR,
3REDUSUR, BLUSUB, REDSUB, PRAIR1, PRAIR2, PRAIR1, PRAIR2, PBSUR1, PBSUR2,
4UPASUR1, PFSUR1, PFSUR2, PBSUB1, PBSUB2, PBSUB1, PBSUB2
DIMENSION ALPHA(12, 71), NAMEWP(29, 61)
1, ‘J’, ’’, ’N ‘)
IF (INC.EQ.0) GO TO 20
WRITE(6, 11)
1 FORMAT (’’)
COUNT=0
DO 10 PLT-AC1, AC2
1 IF (PLATFM(PLT), EQ, 0).AND. (PLATFM(PLT, 24), EQ, 0) GO TO 10
WRITE(6, 41) (NAME(PLT, J), J=1, 6), (NAME(PLATFM(PLT, 21), J), J=1, 6), PLAT
2FM(PLT, 21)
4 FORMAT (’/’, ’AIRCRAFT – ’, 6A1, 5X, ’PARENT SHIP – ’, 6A1, 5X,
1 ’COMBAT LOCATION – ’, 14)
DO 7 WP=1, 2
RESULT=PLATFM(PLT, WP+22)
8 IF (RESULT.EQ.0) GO TO 7
R=RESULT/6.
9 IF (RESULT.GE.1.AND. RESULT.LE.6) WRITE (6, 5) WP, (NAMEWP(PLATFM(PLT, W
1P=16), J), J=1, 6), R
5 FORMAT (’’)
RESULT=RESULT-6.
6 IF (RESULT.GT.6) WRITE (6, 6) WP, (NAMEWP(PLATFM(PLT, WP+16), J), J=1, 6),
7ALPHA1 (RESULT-6), J, J=1, 7
6 FORMAT (’’)
RESULT=RESULT-6.
7 CONTINUE
8 FORMAT (’/’, ’WEAPON ‘, 11, 3X, ’TYPE – ’, 6A2, 5X, ’PROB (HIT)/DETONATION – ’
1, 14)
IF (RESULT.GT.6) WRITE (6, 7) WP, (NAMEWP(PLATFM(PLT, WP+16), J), J=1, 6),
8 ALPHAI (RESULT-6), J, J=1, 7
7 IF (RESULT.LE.6) WRITE (6, 8) WP, (NAMEWP(PLATFM(PLT, WP+16), J), J=1, 6),
9 ALPHAI (RESULT-6), J, J=1, 7
8 CONTINUE
9 CONTINUE
10 WRITE(6, 9)
8 FORMAT (’/’, ’MORE ON NEXT PAGE. TYPE TIES TO CONTINUE.’)
9 READ(5, 9) G
10 FORMAT (’’)
COUNT=0
11 WRITE(6, 10)
12 FORMAT (/”, ”)
13 CONTINUE
14 WRITE(6, 11)
15 FORMAT (/”, ”)
16 CONTINUE

COUNT=0
IF(SUB.EQ.0) GO TO 18
A+SUR1
B+SUR2
C=1
13 DO 16 PLT=A,B
IF((PLATFORM(PLT,35).EQ.0).AND.((PLATFORM(PLT,36).EQ.0).AND.
1(PLATFORM(PLT,37)).EQ.0)) GO TO 16
WRITE(6,14) (NAME(PLT,J),J=1,6),PLATFORM(PLT,2)
14 FORMAT('/1,'SHIP=',G1,A15,5X,'COMBAT LOCATION=',I14)
16 DO 15 M=1,3
RESULT=PLATFORM(PLT,WP+34)
15 IF(RESULT.EQ.0) GO TO 15
A=RESULT/6.
16 IF(RESULT.GE.1.AND.RESULT.LE.6) WRITE(6,5) WP.
65 I(NAMEWP PLATFORM(PLT,WP+20),J),J=1,6),A
66 IF(RESULT.GT.6) WRITE(6,6) WP,(NAMEWP PLATFORM(PLT,WP+20),J),J=1,6),A
67 I(ALPHA):(RESULT-6),J),J=1,7)
15 CONTINUE
70 COUNT=COUNT+1
71 IF(COUNT.LE.4) GO TO 16
72 WRITE(6,8)
73 READ(15,9) G
74 COUNT=0
16 CONTINUE
76 IF(C.GT.11) GO TO 9999
77 18 IF(SUB.EQ.0) GO TO 9999
78 C=2
79 A+SUB1
80 B+SUB2
81 GO TO 13
82 9999 WRITE(6,17)
83 17 FORMAT('END OF WEAPON RESULT DISPLAY. TYPE YES TO CONTINUE.')
84 READ(15,9) G
85 RETURN
86 END
SUBROUTINE DAMRE(SUR1,SUR2,SUB1,SUB2,SUR,SUB)

C THIS SUBROUTINE DISPLAYS DAMAGE SUFFERED BY THE CALLING FORCE

IMPLICIT INTEGER (A-Z)
COMMON PLTFLM(100,37).DETECT(100,4).WEP(29,3).HIT(100,3).SUNK(100)
1.WATER(100).NAME(100,6).BUFFER(100,6).BLUPAS(6).REDPAS(6).
2.PLATM,IRIR,ISURF,ISUB,ADV,POV,REV,ERATIC,BLUAIR,REDAIR,BLUSUR,
3.RESSUR,BLUSUB,RESSUB,PBAIR1,PBAIR2,PRAIR1,PRAIR2,PBSUR1,PBSUR2,
4.PSUR1,PSSHUR2,PSSHUB,PBSSUB2,PBSUB1,PBSUB2
WRITE(6,11)
1 FORMAT('1',5X,'SHIP',5X,'STATUS/HITS')
COUNT=0
IF(SUR.EQ.0) GO TO 9
A=SUR1
B=SUR2
C=1
DO 7 PLT=A.B
IF(PLATFM(PLT,201.EQ.0) WRITE(6,3) (NAME(PLT,J),J=1,6)
7 FORMAT('0',4X,6AI,7X,'SUNK')
TEMP=PLATFM(PLT,11)
IF(PLATFM(PLT,201.GT.0) WRITE(6,4) (NAME(PLT,J),J=1,6),TEMP
4 FORMAT('0',4X,6AI,9X,11)
COUNT=COUNT+1
IF(COUNT.LT.10) GO TO 7
COUNT=0
WRITE(6,5) GO TO 2
5 FORMAT('MORE ON NEXT PAGE. TYPE YES TO CONTINUE.')
READ(5,6) (NAME(PLT,J),J=1,6)
6 FORMAT('
7 CONTINUE
9 IF(C.GT.1) GO TO 9999
9 IF(SUB.EQ.0) GO TO 9999
C=2
A=SUB1
B=SUB2
GO TO 2
9999 WRITE(6,8)
5 FORMAT('END OF DAMAGE DISPLAY ROUTINE. TYPE YES TO CONTINUE.')
READ(5,6)
RETURN
END
SUBROUTINE STAT(RCI,AC2,SUI,SUR2,SUB1,SUB2,NMWPI)
C     DISPLAYS INDIVIDUAL PLATFORM STATUS OF COMBAT ACTIONS, MOVEMENT
C     AND SEARCH RESULTS. ACCEPTS COMBAT, MOVEMENT AND SEARCH
C     COMMANDS. ALL INPUT IS PUT INTO A BUFFER AND OVERLAYED AT
C     GAME TURN EXECUTION

IMPLICIT INTEGER(A-Z)
COMMON PLATFORM(100,37),DETECT(100,4),WEP(29,3),HIT(100,3),SUNK(100)
      WATER(100),NAME(100,6),BUFFER(100,6),BLUPAS(6),REDPAS(6),
      2PLAT,ISAAP,ISUB,ADV,ADV,PEV,ERATIC,BLUAIR,REDAIR,BLUSUR,
      3REDUR,BLUSUB,REDSUB,P5AI2,P5AIR2,P5AIR3,P5AIR4,P5SUR1,P5SUR2,
      4P5SUR3,P5SUR4,P5SUR5,P5SUR6,P5SUR7,P5SUR8
DIMENSION INPUT (70),INAME (61),NAMWPI (29,6),EXIT (6)
DATA I0/'N'/
      WRITE(6,100)
FORMAT (III'YOU HAVE SELECTED INDIVIDUAL PLATFORM STATUS DISPLAY',
      1' AND ASSOCIATED'' COMBAT, MOVEMENT AND SEARCH COMMAND INPUTS.'/
      2' ENTER THE NAME OF THE DESIRED PLATFORM.'/
      READ (5,101)INPUT
      FORMAT(70A1)
      CALL INVAT6(INPUT,NNAME)
      CALL NAMEIN(INAME,INDEX,NRME,AC1,AC2,SUI,SUR,SUB,SB2)
      IF (INDEX.NE.00) GO TO 20
      WRITE(6,102)
      FORMAT (III'YOU HAVE ENTERED AN INCORRECT PLATFORM NAME.'/
      1' 'REENTER 1'/)
      GO TO 10
      TEST=0
      IF (INDEX.GE.AC1.AND.INDEX.LE.AC2) CALL STATAC(TEST,INDEX,NNAME)
      IF (INDEX.GE.SUI.AND.INDEX.LE.SUR2) CALL STATSS(TEST,INDEX,NNAME)
      IF (INDEX.GE.SUB1.AND.INDEX.LE.SUB2) CALL STATBS(TEST,INDEX,NNAME)
      IF (TEST.EQ.0) WRITE (6,103)
      FORMAT (III'YOU CAN ONLY OBTAIN STATUS AND INPUT COMMANDS'/
      1' FOR PLATFORMS UNDER YOUR CONTROL.'/
      1' FOR PLATFORMS UNDER YOUR CONTROL.'/
      1' FOR PLATFORMS UNDER YOUR CONTROL.'/
      1' FOR PLATFORMS UNDER YOUR CONTROL.'/
      WRITE (6,104)
      FORMAT (III'ENTER YES) TO CONTINUE WITH ANOTHER PLATFORM'/
      1' ENTER NO) TO RETURN TO DISPLAY SELECTION'
      READ (6,105) INPUT
      IF (TEST.EQ.0) RETURN
      WRITE (6,106)
      FORMAT (III' ENTER THE NAME OF THE DESIRED PLATFORM'/)
      GO TO 10
END
SUBROUTINE STATAC(USED, INDEX, INAME, NAME WP)

C

AIRCRAFT STATUS AND COMMAND INPUT SUBROUTINE

C

IMPLICIT INTEGER(A-Z)

COMMON PLATFM(100,7), DETECT(100,4), WEP (29,3), HIT (100,3), SUNK (100)

1, WATERR100), NAME (100, 6), BUFFER (100, 6), BLUPAS (6), REDPAS (6),

2IPLAT, IRC, ISUB, RDI, DOW, REV, FRIC, BLURIR, REDAIR, BLUSUB,

3REDURA, BLUSUB, REDSUB, PBAIR1, PBAIR2, PBAIR1, PBAIR2, PBSUR1, PBSUR2,

4PASUR1, PASUR2, PBSUR1, PBSUR2, PBSUR2

DIMENSION INAME (6), NAMEWP (29, 6), INPUT (70), WEPON (6)

DATA NONE/'N'/

15

USED=1

16 IF (PLATFM(INDEX, 11).EQ.0) WRITE (6, 100) NAME

17 100 FORMAT (//IX,6A1, 

18 ' IS CURRENTLY NOT FLYING - IT IS ASSUMED YOU'

19 1' DESIRE TO LAUNCH. ')

20 IF (PLATFM(INDEX, 11).EQ.0) GO TO 35

21

C

22 STATUS DISPLAYS

23

24 WRITE (6, 101) INAME, PLATFM(INDEX, 1), PLATFM(INDEX, 4)

25 101 FORMAT (IX,6A1, ' IS LOCATED AT POSITION ', IS, ' AT SPEED ', IS)

26 IF (PLATFM(INDEX, 23).EQ.0 .AND. PLATFM(INDEX, 24).EQ.0) GO TO 10

27 X=23

28 IF (PLATFM(INDEX, 23).EQ.0) X=24

29 WRITE (6, 102) NAMEWP (PLATFM(INDEX, X-6), J), J=1, 6, PLATFM(INDEX, 2)

30 102 FORMAT (IT RELEASED A ', 6A2, ' AT A SUBMARINE TARGET LOCATED /

31 1' AT POSITION ', IS, ' THE WEAPON ')

32 IF (PLATFM(INDEX, X).LE.6) WRITE (6, 103) PLATFM(INDEX, X)

33 103 FORMAT (IT DETONATED WITH PROBABILITY OF A HIT OF ', IS, 1

34 1' OUT OF 6 ')

35 IF (PLATFM(INDEX, X).EQ.7) WRITE (6, 104)

36 104 FORMAT (IT DETONATED (HIT ' 1)

37 IF (PLATFM(INDEX, X).EQ.8) WRITE (6, 105)

38 105 FORMAT (IT ' DID NOT DETONATE ')

39 IF (PLATFM(INDEX, 7).EQ.0) WRITE (6, 106)

40 106 FORMAT (IT WAS NOT ASSIGNED A SEARCH MISSION ')

41 IF (PLATFM(INDEX, 7).EQ.0) GO TO 30

42 IF (PLATFM(INDEX, 4).GT.0) WRITE (6, 107) INAME

43 107 FORMAT (IT COULD NOT EXECUTE A SEARCH MISSION AS ITS SPEED ')

44 1' WAS GREATER THAN ZERO ')

45 IF (PLATFM(INDEX, 4).GT.0) GO TO 30

46 IF (PLATFM(INDEX, 7).EQ.1) WRITE (6, 108)

47 108 FORMAT (IT CONDUCTED AN ACTIVE SEARCH WITH FOLLOWING RESULTS ')

48 1' SUBMARINE CONTACT LOCATION RANGE ')

49 IF (PLATFM(INDEX, 7).EQ.2) WRITE (6, 109)

50
FORMAT(' IT CONDUCTED A PASSIVE SEARCH WITH FOLLOWING RESULTS'/*
1' SUBMARINE CONTACT LOCATION RANGE')

51 109 FORMAT(' IT CONDUCTED A PASSIVE SEARCH WITH FOLLOWING RESULTS'/*
52 1' SUBMARINE CONTACT LOCATION RANGE')
53 X=0
54 DO 20 I=1,100
55 IF(DETECT(I,1) LT .01) GO TO 2
56 IF(DETECT(I,1) NE INDEX) GO TO 20
57 X=1
58 WRITE (6,110) PLATFORM(DETECT(I,2)),DETECT(I,4)
59 110 FORMAT('T13,14,31,14')
60 20 CONTINUE
61 21 IF(X.EQ.0) WRITE(6,111)
62 111 FORMAT(*, 'THERE WERE NO DETECTIONS')
63 WRITE (6,127) NAME(INDEX,J),J=1,6
64 127 FORMAT(*, 'WEAPON SYSTEM(S)'/*TS,'TYPE','TIS,'RANGE')
65 WRITE (6,128) NAMEWP(PLATFORM(INDEX,17),J),J=1,6),WEP(PLATFORM(INDEX,
66 117),2)
67 128 FORMAT('1' , '6A',13)
68 IF(PLATFORM(INDEX,18).EQ.0) GO TO 30
69 WRITE (6,129) NAMEWP(PLATFORM(INDEX,18),J),J=1,6),WEP(PLATFORM(INDEX,
70 119),2)
71 129 FORMAT('2' ',6A',13)
72 C
73 C COMMAND INPUTS
74 C
75 C
76 C
77 30 WRITE(6,112)
78 112 FORMAT('NO COMBAT ENTER N(ONE) OTHERWISE ENTER'/*
79 '1' TARGET LOCATION'/*WEAPON SYSTEM NUMBER IN THIS FASHION'/*')
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 113 FORMAT('70A')
88 CALL INVRT1(INPUT,TEST)
89 IF(1.EQ.0) GO TO 40
90 CALL NONVRT(INPUT,BUFFER(INDEX,2))
91 READ(5,113) INPUT
92 CALL NONVRT(INPUT,BUFFER(INDEX,1))
93 GO TO 40
94 C
95 C
96 C
97 C
98 C
99 35 WRITE(6,114) NAME
100 114 FORMAT('IX,6A1,' IS NOW BEING LAUNCHED')
PLATFM(INDEX,11)=1

MOVE

107 40 WRITE(6,115)
108 115 FORMAT('MOVEMENT NOT DESIRED ENTER NONE OTHERWISE ENTER'/
109 1 'SPEED'/ 'DESTINATION HEX IN THIS FASHION'/)
110 READ(5,113) INPUT
111 CALL INVRTI(INPUT,TEST)
112 IF (TEST.EQ.NONE) BUFFER(1,4)=PLATFM(1,3)
113 IF (TEST.EQ.NONE) GO TO 50
114 CALL NONVRT(INPUT,BUFFER(INDEX,3))
115 READ(5,113) INPUT
116 CALL NONVRT(INPUT,BUFFER(INDEX,4))
117 TEST FOR GOING RIGHT DISTANCE AND NOT EXCEEDING MAX SPEED
118 CALL DISTAN(BUFFER(INDEX,4),PLATFM(INDEX,3),DIST)
119 IF (DIST.LE.BUFFER(INDEX,3).AND.BUFFER(INDEX,3).LE.PLATFM(INDEX,12)
120 ) GO TO 50
121 WRITE(6,116)
122 FORMAT('YOU HAVE EITHER SET A SPEED HIGHER THAN MAX'/
123 'OR ATTEMPTED TO GO FURTHER THAN SET SPEED'/)
124 GO TO 40
125 SEARCH
126 WRITE(6,117)
127 FORMAT('ENTER SEARCH MISSIONS AS FOLLOWS: '/T7,'O NONE SCHEDULED
128 '1' ACTIVE MISSION'/T7,'2 PASSIVE MISSION'/)
129 READ(5,113) INPUT
130 CALL NONVRT(INPUT,X)
131 IF (X.GT.0.AND.BUFFER(INDEX,3).GT.0) WRITE(6,118)
132 FORMAT('A SEARCH MISSION CANNOT BE PERFORMED WHILE THIS PLATFORM/
133 IS MOVING (NOT HOVERING OR LOITERING)/')
134 IF (X.GT.0.AND.BUFFER(INDEX,3).GT.0) GO TO 50
135 BUFFER(INDEX,5)=X
136 COMMAND STATUS/INPUT COMPLETION
137 WRITE(6,119)
138 FORMAT('YOU HAVE ENTERED THE FOLLOWING COMMANDS'/
139 'COMBAT',')
151 IF(BU弗ER(INDEX,1).EQ.0) WRITE (6,120)
152 FORMAT(' NONE SCHEDULED')
153 IF(BU弗ER(INDEX,1).GT.0) WRITE (6,121) NAMEibbon Platm(INDEX,BU弗ER(IN
154 INDEX,1)*15),J, J=1,61, BUFFER(INDEX,2)
155 FORMAT('X,6A2,' WEPON TGT LOCATION',15)
156 IF(BU弗ER(INDEX,3).EQ.0) WRITE (6,122) Platm(INDEX,3)
157 FORMAT(' SPEED IS ZERO AT LOCATION ',14)
158 IF(BU弗ER(INDEX,3).GT.0) WRITE (6,123) BUFFER(INDEX,4), BUFFER(INDEX,13)
159 123 FORMAT(' DESTINATION',15,' SPEED',13)
160 IF(BU弗ER(INDEX,5).EQ.0) WRITE (6,124)
161 124 FORMAT(' THERE IS NO SEARCH MISSION SCHEDULED')
162 IF(BU弗ER(INDEX,5).EQ.1) WRITE (6,125)
163 125 FORMAT(' AN ACTIVE SEARCH MISSION WILL BE CONDUCTED')
164 IF(BU弗ER(INDEX,5).EQ.2) WRITE (6,126)
165 126 FORMAT(' A PASSIVE SEARCH MISSION WILL BE CONDUCTED')
166 RETURN
167 END
SUBROUTINE STATSSUSEI(INDEX, INAME, NAMEWP)

DISPLAYS INDIVIDUAL SURFACE SHIP/SUBMARINE STATUS OF COMBAT ACTIONS, MOVEMENT AND SEARCH RESULTS. ACCEPTS COMBAT MOVEMENT AND SEARCH COMMANDS. ALL INPUT IS PLACED IN A BUFFER AND OVERLAYERED AT GAME TURN EXECUTION

IMPLICIT INTEGER(A-Z)

COMMON PLATFM(100,37), DETECT (100,29), WEP (29,31), HIT (100,31), SUNK (100,31)

DIMENSION INAME (6), NAMEWP (29,6), INPUT (70)

DATA NONE/*N*/ USeD=1

STATUS DISPLAYS

WRITE (6,100) INAME, PLATFM(INDEX,31), PLATFM(INDEX,4)

100 FORMAT (/IX,6AI," IS LOCATED AT POSITION",IS," SPEED",12)

IF(PLATFM(INDEX,11).LT.O) WRITE (6,99) PLATFM(INDEX,12), TEMP

99 FORMAT (/T10,"NOTE:" DUE TO PREVIOUS AND/OR CURRENT DAMAGE SUFFERED MAX SPEED IS NOW",12," THE SHIP HAS BEEN HIT",12," TIMES")

WRITE (6,101) INDEX, NAMEWP

101 FORMAT (' IT HAS BEEN HIT BY A NON-TORPEDO WEAPON')

WRITE (6,102) INDEX, NAMEWP

102 FORMAT (' IT HAS BEEN HIT BY A TORPEDO')

CONTINUE

COMBAT ACTION RESULTS

IF(PLATFM(INDEX,11).EQ.0) GO TO 30

WRITE (6,103) INAME, PLATFM(INDEX,2)

103 FORMAT (/IS," THE ",6AI," ATTACK ON TARGET LOCATED AT",15/I)
DO 20 I=35,37
IF (PLATFM(INDEX,1).EQ.0) GO TO 20
WRITE (6,104) (NAMEW(PLATFM(INDEX,1-14),J),J=1,6)
104 FORMAT('RELEASE OF ',6A2,' RESULTED IN')
IF (PLATFM(INDEX,1).LE.6) WRITE (6,105) PLATFM(INDEX,1)
105 FORMAT('A DETONATION WITH PROBABILITY/" OF A HIT OF "/
111," OUT OF 6")
IF (PLATFM(INDEX,1).EQ.7) WRITE (6,106)
106 FORMAT('A DETONATION HIT/')
IF (PLATFM(INDEX,1).EQ.8) WRITE (6,107)
107 FORMAT('NO DETONATION - NO DISTURBED WATER MARKER/
117 CONTINUE
C
C SEARCH MISSION STATUS
C
30 IF (PLATFM(INDEX,7).EQ.0) WRITE (6,108)
108 FORMAT('IT WAS NOT ASSIGNED A SEARCH MISSION')
IF (PLATFM(INDEX,7).EQ.0) GO TO 50
WRITE (6,109)
109 FORMAT('IT WAS ASSIGNED THE FOLLOWING SEARCH MISSION(S):')
WRITE (6,110)
110 FORMAT('ACTIVE')
IF (PLATFM(INDEX,7).EQ.1) WRITE (6,111)
111 FORMAT('PASSIVE')
IF (PLATFM(INDEX,7).EQ.3) WRITE (6,112)
112 FORMAT('ACTIVE AND PASSIVE')
WRITE (6,113)
113 FORMAT('TYPE LOCATION METHOD RANGE')
X=0
DO 40 I=1,100
40 IF (DETECT(I,1).LT.0) GO TO 41
IF (DETECT(I,1).NE.INDEX) GO TO 40
X=1
IF (PLATFM(DETECT(I,2),1).EQ.0) WRITE (6,114)
114 FORMAT('SURFACE',5X,14.5X,'PASSIVE',16)
IF (PLATFM(DETECT(I,2),1).EQ.2) WRITE (6,115)
115 FORMAT('SUBMARINE',5X,14.5X,'ACTIVE',17)
IF (X.EQ.0) WRITE (6,116)
116 FORMAT('THERE WERE NO DETECTIONS')
IF SUNK - SKIP COMMAND INPUTS

IF (PLATFM(INDEX,20).EQ.0) WRITE (6,117) NAME
"I" CAN BE ACCEPTED'"
IF (PLATFM(INDEX,20).EQ.0) RETURN
WRITE (6,134) (NAME(INDEX,J),J=1,6)
FORMAT (1x,6A1, ' WEAPON SYSTEM IS', 'TYPE', 'TS', 'RANGE')
DO 51 I=1,3
11*1=20
IF(PLATFM(INDEX,11).EQ.0) GO TO 52
WRITE (6,135) (INDEX,11), (INDEX,J), (INDEX,J), (INDEX,J)
WRITE (6,120)
117 FORMAT (12,2X,6A2,13)
118 51 CONTINUE
119 C
120 C
121 C
122 C
123 C
124 52 WRITE(6,118)
125 118 FORMAT(1X,' NO COMBAT ENTER N(ONE) - OTHERWISE ENTER */' TARGET Loca
126 ' NITION*/ ' WEAPON SYSTEM NUMBER IN THIS FASHION*/'
127 READ(5,119) INPUT
128 119 C
129 CALL INVATI(INPUT,TEST)
130 IF (TEST.EQ.0) GO TO 70
131 CALL NONVRT(INPUT,BUFFER(INDEX,2))
132 READ(5,119) INPUT
133 60 CALL NONVRT(INPUT,WEAPON)
134 BUFFER(INDEX,11)=
135 BUFFER(INDEX,WEAPON+51)=
136 WRITE(6,120)
137 120 FORMAT(1X,' TO EMPLOY ANOTHER WEAPON SYSTEM, ENTER THE SYSTEM NUMBER'
138 ' OTHERWISE ENTER N(ONE) */'
139 READ(5,119) INPUT
140 CALL INVATI(INPUT,TEST)
141 IF (TEST.EQ.0) GO TO 70
142 GO TO 60
143 C
144 C
145 C
146 C
147 C
148 70 WRITE(6,121)
149 121 FORMAT(1X,' TO SCHEDULE MOVEMENT ENTER */ SPEED */ DESTINATION
150 1 IN THAT ORDER, OTHERWISE ENTER N(ONE) */

218
READ(5,119)INPUT
CALL INVRT(INPUT,TEST)
C IF(TEST.EQ.NONE)BUFFER(INDEX,4)=PLATFM(INDEX,3)
C IF(TEST.EQ.NONE)GO TO 80
CALL NONVRT(INPUT,BUFFER(INDEX,3))
READ(5,119)INPUT
CALL NONVRT(INPUT,BUFFER(INDEX,4))

C TEST FOR GOING RIGHT DISTANCE AND NOT EXCEEDING MAX SPEED
C
CALL DISTAN(BUFFER(INDEX,4),PLATFM(INDEX,3),DIST)
IF(DIST.LE.BUFFER(INDEX,3).AND.BUFFER(INDEX,3).LE.PLATFM(INDEX,12))
1)GO TO 80
WRITE(6,122)
122 FORMAT(' YOU HAVE EITHER SET A SPEED HIGHER THAN MAX CAPABLE OR AT
ITEMPTED/' TO GO FURTHER THAN THE SPEED SETTING WILL ALLOW')
GO TO 70
C
C SEARCH COMMANDS
C
WRITE(6,123)
123 FORMAT(' ENTER SEARCH MISSIONS AS FOLLOWS:'/'T7.'0 NONE SCHEDULE
1D.'T7.'1 ACTIVE MISSION'/'T7.'2 PASSIVE MISSION'/'T7.'3 ACTIVE AN
20 PASSIVE'/'
READ(5,119)INPUT
C CALL NONVRT(INPUT,X)
IF(X.EQ.3.AND.PLATFM(INDEX,24).NE.1)WRITE(6,124)
124 FORMAT(' AN ACTIVE AND PASSIVE SEARCH MISSION CANNOT BE PERFORMED
1 SIMULTANEOUSLY' BY THIS PLATFORM')
IF(X.EQ.3.AND.PLATFM(INDEX,24).NE.1)GO TO 80
BUFFER(INDEX,5)=X
C
C STATUS/COMMAND INPUT COMPLETION
C
WRITE(6,125)
125 FORMAT(' YOU HAVE ENTERED THE FOLLOWING COMMANDS'/' COMBAT:'
C IF(BUFFER(INDEX,1).EQ.0)WRITE(6,126)
126 FORMAT(' NOT SCHEDULED')
IF(BUFFER(INDEX,1).EQ.0)GO TO 91
DO 90 I=6,8
IF(BUFFER(INDEX,1).GE.1)GO TO 90
WRITE(6,127)NAME,PLATFM(INDEX,1+IS),J1,J1,M1,61,BUFFER(INDEX,2)
91 90 CONTINUE
201  IF (BUFFER(INDEX, 3) .EQ. 0) WRITE (6, 128) PLATFM (INDEX, 3)
202  FORMT ('SPEED ZERO LOCATION', 15)
203  IF (BUFFER(INDEX, 3) .GT. 0) WRITE (6, 129) BUFFER(INDEX, 4), BUFFER(INDEX, 3)
204  IF (X.EQ.0) WRITE (6, 120)
205  FORMT ('DESTINATION ', 14, ' SPEED', 12)
206  X=BUFFER(INDEX, 5)
207  IF (X.EQ.0) WRITE (6, 121)
208  FORMAT ('NO SEARCH MISSION')
209  IF (X.EQ.1) WRITE (6, 131)
210  FORMAT ('ACTIVE SEARCH MISSION')
211  IF (X.EQ.2) WRITE (6, 132)
212  FORMAT ('PASSIVE SEARCH MISSION')
213  IF (X.EQ.3) WRITE (6, 133)
214  FORMAT ('ACTIVE AND PASSIVE SEARCH MISSION')
215  RETURN
216  END
SUBROUTINE CONTAL

THIS SUBROUTINE EXECUTES A GAME TURN OF COMBAT, MOVE AND SEARCH

IMPLICIT INTEGER (A-M,P-Z)

COMMON PLATFORM(100,37), DETECT(100,4), WEP(29,3), HIT(100,3), SUNK(100)
1, WATER(100), NAME(100,6), BUFFER(100,8), BLUPAS(8), REDPAS(8),
2, PLAT, IAIR, ISURF, ISUB, ADV, PDY, REV, REV2, ERATIC, BLUAIR, REDAIR, BLUSUR,
3, REDSUR, BLUSUR, NEOSUB, PBAIR1, PBAIR2, PPAIR1, PPAIR2, POSUR1, POSUR2,
4, NPASUR1, NPASUR2, NPSUR1, NPSUR2, NPASUR1, NPASUR2

WORKING MATRIX INITIALIZATION

DO 10 I=1,100
10 WATER(I)=1
DO 20 I=1,11 PLATM(I)=0
20 CONTINUE
IF (IAIR.EQ.0) GO TO 21
DO 21 I=1,IAIR
21 CONTINUE
IF (IAIR.EQ.0) GO TO 21
DO 22 I=1,1 PLATM(I)=0
22 CONTINUE
IF (IAIR.EQ.0) GO TO 21
DO 23 I=1,IAIR
23 CONTINUE
OVERLAY BUFFER INTO PLATFORM

MOVES NONFLYING AIRCRAFT WITH MOTHER SHIP

IF (IAIR.EQ.0) GO TO 24
DO 24 I=1,IAIR
24 CONTINUE
OVERLAY BUFFER INTO PLATFORM
DO 25 I=1,1PLAT

PLATM(1,1)=BUFFER(1,1)
PLATM(1,2)=BUFFER(1,2)
PLATM(1,4)=BUFFER(1,3)
PLATM(1,5)=BUFFER(1,4)
PLATM(1,7)=BUFFER(1,5)
IF(I.LT.1)GO TO 25
PLATM(1,21)=PLATM(1,21)=BUFFER(1,6)
PLATM(1,22)=PLATM(1,22)=BUFFER(1,7)
PLATM(1,23)=PLATM(1,23)=BUFFER(1,8)
CONTINUE

COMBAT AND MOVE EXECUTION

CALL COMBAT
CALL MOVE

INITIALIZATION OF DETECT MATRIX

DO 30 J=1,100
DETECT(I,1)=1
CONTINUE

PLACE DISTURBED WATER MARKERS IN PLATFORM MATRIX

NUM=0
DO 40 I=1,100
IF(WATER(I).LT.1)GO TO 50
NUM=NUM+1
CONTINUE
50 IF(NUM.EQ.0)GO TO 80
DO 70 I=1,1PLAT
IF(PLATM(I,6).EQ.3.OR.PLATM(I,20).EQ.0)GO TO 70
DO 60 J=1,NUM
CALL DISTANIPLATM(I,3),WATER(I,J),DIST
IF(DIST.LE.11)PLATM(I,6)=3
CONTINUE
60 CONTINUE
70 CONTINUE

EXECUTE SEARCH PHASE
CALL SEARCH

INITIALIZATION OF INPUT COMMAND BUFFER

DO 100 I=1,1PLAT
BUFFER(I,1)=0
BUFFER(I,2)=0
BUFFER(I,3)=0
BUFFER(I,4)=PLATFORM(I,3)
BUFFER(I,5)=0
BUFFER(I,6)=1
BUFFER(I,7)=1
BUFFER(I,8)=1
CONTINUE
RETURN
END
SUBROUTINE COMBAT

THIS SUBROUTINE ACTIVATES THE MASTER COMBAT PHASE AND CONTROLS ALL COMBAT RELATED SUBROUTINES

IMPLICIT INTEGER (A-H, P-Z)
COMMON PLATFORM, DETECT, WEP, HIT, SUNK
I WATER, NAME, BUFFER, BLUPAS, REDPAS
2 PLAT, WEP, WEPSUP, HIT, ITAR, PBAIR, PBAIR2, PBSUR, PBSUR2,
3 WPSUR, WEPSUB, PBSUR, PBSUR2, PBAIR, PBAIR2, PBSUR, PBSUR2,
4 DO 30 ATTK=1 PLATFORM
5 IF (PLATFORM.NE.0) GO TO 30
6 IF (PLATFORM.EQ.0) GO TO 30
7 IF (PLATFORM.EQ.0) GO TO 10
8 WEPIND=16+PLATFORM(11)
9 WEPSUP=14+PLATFORM(11)
10 CALL FIRE(ATT, WEPIND, WEPSUP, HIT, ITAR, PAHIT)
11 X=PLATFORM(ATT, WEPIND)
12 IF (X .LT. 4. OR. X .GT. 19) PLATFORM(ATT, WEPIND) = PAHIT
13 IF (X.GE.4. AND. X.LE.19. AND. HIT.EQ.1) PLATFORM(ATT, WEPIND) = 7
14 IF (X.GE.4. AND. X.LE.19. AND. HIT.EQ.0) PLATFORM(ATT, WEPIND) = 8
15 IF (HIT.EQ.1) CALL FILM ITI ATTK, PLATFORM(ATT, WEPIND), ITAR
16 GO TO 30
17 CONTINUE

CONTROLS SURFACE AND SUBSURFACE WEAPON EMPLOYMENT

DO 20 WEPIND=21, 29
20 IF (PLATFORM(ATT, WEPIND).GE.0) GO TO 20
21 PLATFORM(ATT, WEPIND) = PLATFORM(ATT, WEPIND)
22 WEPSUP=WEPIND-6
23 CALL FIRE(ATT, WEPIND, WEPSUP, HIT, ITAR, PAHIT)
24 X=PLATFORM(ATT, WEPIND)
25 IF (X .LT. 4. OR. X .GT. 19) PLATFORM(ATT, WEPIND) = PAHIT
26 IF (X.GE.4. AND. X.LE.19. AND. HIT.EQ.1) PLATFORM(ATT, WEPIND) = 7
27 IF (X.GE.4. AND. X.LE.19. AND. HIT.EQ.0) PLATFORM(ATT, WEPIND) = 8
28 IF (HIT.EQ.1) CALL FILM ITI ATTK, PLATFORM(ATT, WEPIND), ITAR
29 CONTINUE
30 CONTINUE

224
51 C COMBAT EXECUTION OVER - DAMAGE ASSESSMENT PHASE
52 C
53 C
54 35 DO 40 I=1,100
55 IF(INIT(I,1).LE.0) GO TO 50
56 CALL DAMREC(INIT(I,1))
57 40 CONTINUE
58 WRITE (6,200)
59 200 FORMAT(1' THERE WERE GREATER THAN 100 HITS >100 HITS NOT FILED')
60 C
61 C
62 C COMPRESS DISTURBED WATER MATRIX
63 C
64 C
65 C
66 50 I=0
67 DO 60 J=I+1
68 IF(WATER(I,J).LT.0) GO TO 100
69 J=J+1
70 CONTINUE
71 IF(WATER(I,J).LT.0) GO TO 60
72 DO 80 K=J,100
73 IF(WATER(K,J).LT.0) GO TO 80
74 WATER(K,J)=WATER(K,J+1)
75 CONTINUE
76 DO 90 K=J,100
77 IF(K.GE.100) GO TO 100
78 90 CONTINUE
79 100 RETURN
80 END
SUBROUTINE FIRE(ATTK, WEPIND, WPLEFT, HIT, TAR, PHITI)

THIS SUBROUTINE CONDUCTS ALL FIRINGS, CALCULATES
PROBABILITY OF HIT, DRAWS A RANDOM NUMBER AND
DETERMINES HIT OR MISS

IMPLICIT INTEGER(A-2)
REAL RN
COMMON PLATFM(100, 37), DETECT(100, 4), WEP(29, 3), HIT(100, 3), SUNK(100)
WATER(100), NAME(100, 6), BUFFER(100, 8), BLUPAS(6), RDPAS(6),
2IPAT, TITA, SURF, SUB, ADV, REV, PEV, ERATIC, BLUR, REDAIR, BLUSUR,
3REDUR, BLUSUB, RDAIR, RLAIR, RLAIR2, PBAIR, PBAIR2, PBSUR, PBSUR2,
4PASUR, PASUR2, PBSUB, PBSUB2, PBSUB1, PBSUB1, PBSUB2
DATA ISLED/55555/
1 INIT=0
IF (PLATFM(ATTK, WPLEFT) .EQ. 0) GO TO 30
CHECK IF COMBAT LOCATION IS A TARGET PLATFORM
PLATFM(ATTK, WPLEFT) = PLATFM(ATTK, WPLEFT) - 1
ITAR=0
DO 10 INDEX=1, IPLAT
IF (PLATFM(INDEX, 20).EQ.0) GO TO 10
IF (PLATFM(INDEX, 21).EQ. PLATFM(INDEX, 31)) ITAR=INDEX
CONTINUE
IF (ITAR.EQ.0) GO TO 30
PLACE DISTURBED WATER MARKER AT TGT LOCATION
IF WEAPON IS NOT A TORPEDO
IF (WEP(WEPIND, 3).EQ.3) GO TO 14
CALL DISTUR(PLATFMIITAR, 51)
CHECK IF TARGET PREVIOUSLY DETECTED
DO 15 I=1, 100
IF (DETECT(I, 1).LT.0) GO TO 16
IF (DETECT(I, 1).EQ. ATTK AND DETECT(I, 2).EQ. ITAR) Y=1
CONTINUE
IF (Y.EQ.0) GO TO 30
226
51  C
52  C  CHECK IF PROPER WEAPON AGAINST PARTICULAR TARGET
53  C
54  C
55      T=TAIR+ISURF
56      IF (PLATFM(ATTK,1),WEPI,LE,2.AND.ITAR.GT.TI)GO TO 30
57      IF (PLATFM(ATTK,1),WEPI,GE,15.AND.ITAR.LE.TI)GO TO 30
58      IF (ATTK.LE,1.AND.ITAR.GT.TI)GO TO 20
59      IF (ATTK.GT,1.AND.ITAR.GT.TI)GO TO 20
60      RETURN
61  C
62  C  COMPUTE RANGE OF WEAPON AGAINST TARGET RANGE
63  C  COMPUTE PROB OF HIT AND DRAW RANDOM NR
64  C  DETERMINE HIT OR MISS
65  C
66  C
67  C
68  20  CALL DISTANCE(PLATFM(ATTK,2),PLATFM(ATTK,3),DIST)
69  CALL RANGE(PLATFM(ATTK,WEPI),ACC,RANGI)
70      IF (DIST.GT.RANGI)GO TO 40
71      PRHIT=ACC-PLATFM(ITAR,1)
72      IF (ACC.EQ.3)PRHIT=1
73      CALL RANDOM(15,SEED,AN,1)
74      IRN=6.*AN+1.
75      IF (IRN-IRN.GE.0)IRN=1
76  C
77  C  IF WEAPON IS A TORPEDO AND A HIT PLACE DISTURBED
78  C  WATER MARKER AT TGT LOCATION
79  C
80  C
81  C
82      IF (WEP(WEPIND,3).EQ.3.AND.INIT,EQ.1)CALL DISTANCE(PLATFM(ATTK,5))
83      30  RETURN
84      40  WRITE(6,300)NAME(ATTK,J),J=1,6
85      300  FORMAT(' THE ATTACK BY ',6A1,' FAILED AS TGT RANGE > WEAPON RANGE'
86      1)
87      RETURN
88  END
SUBROUTINE DISTURBLOCAT)

IMPLICIT INTEGER (A-H,P-Z)

COMMON PIAFNI(100,37),DETECT(100,4),WEP(29,3),HIT(100,3),SUNK(100)

IF I.WATER(I-LOCRT).LT.0 GO TO 20

RETURN

END
SUBROUTINE RANGE(INDEX, ACC, RANG)

C

C RETRIEVES ACCURACY AND RANGE INFORMATION
C FOR SPECIFIC WEAPON EMPLOYMENT
C

C

IMPLICIT INTEGER(A-Z)

COMMON PLATFORM(100,37), DETECT(100,41), WEP(29,31), HIT(100,31), SUNK(100)
WATER(100), NAME(100,6), BUFFER(100,8), BUPAS(6), REDPAS(6),
2IPLAT, IAIR, ISURF, ISUB, ADV, PDV, REV, PVE, ERATIC, BLUAIR, REDAIR, BLUSUR,
3SURDS, BLUSUB, REDSUB, PBAIR1, PBAIR2, PBAIR1, PBAIR2, PSSUR1, PSSUR2,
4PSSUR1, PSSUR2, PBSUB1, PBSUB2, PBSUB1, PBSUB2

ACC=WEP(INDEX, 1)
RANG=WEP(INDEX, 2)
RETURN
END
SUBROUTINE FILHIT(ATTK, WEPON, ITAR)

C FILES ALL HITS ON SHIPS AS THEY OCCUR

C IMPLICIT INTEGER(A-Z)

COMMON PLATFORM(100,37), DETECT(100,4), WEP(29,31), HIT(100,31), SUNK(100)
1. WATER(100), NAME(100,6), BUFFER(100,8), BLUPAS(6), REDPAS(6),
21 PIAIR, ISURF, ISUB, ADV, PDV, XEV, ERATIC, BLAIR, REDAIR, BLUSUR,
3 REDSUR, BLUSUB, REDSUB, PAIR1, PAIR2, PAIR1, PAIR2, PBSUR1, PBSUR2,
4 PBSUR3, PBSUR4, PBSUB1, PBSUB2, PBSUB3, PBSUB4
5 DO 20 J=1,100
6 IF (HIT(I,J,1).LT.0) GO TO 30
7 CONTINUE
8 WRITE (6,100)
9 FORMAT ('1 HIT DIMENSIONING EXCEEDED-HIT NOT FILED BUT RECORDED')
10 RETURN
11 30 HIT(I,J,1)=ITAR
12 HIT(I,J,2)=WEPON
13 HIT(I,J,3)=ATTK
14 RETURN
15 END
SUBROUTINE DAMRECITARI

       UPDATES ALL DAMAGE PUTS SUNK SHIPS IN SUNK ARRAY

       IMPLICIT INTEGER(A-Z)

       COMMON PLATFM(100,371),DETECT(100,4),WEP(129,3),HIT(100,3),SUNK(100)
       1.WATER(100),NAME(100,6),BUFFER(100,8),BLUPAS(6),REDPAS(6),
       2.IPLAT,ITAR,ISURF,ISUB,ADV,POV,REV,ERATIC,BLUAIR,REDAIR,BLUSUR,
       3.REDSUR,BLUSUR,REDSUB,BLURA1,PBAIR2,PRAIR2,PBSUR1,PBSUR2,
       4.PASUR1,PASUR2,PBSUB1,PBSUB2,PASUB1,PASUB2

       DO 10 J=12,17
       PLATFM(ITAR,J)=PLATFM(ITAR,J)-1
       10 CONTINUE

       IF(PLATFM(ITAR,J)*PLATFM(ITAR,20).LT.0)GO TO 20
       IF(PLATFM(ITAR,20).EQ.0)GO TO 40

       CAPABILITY REDUCTION FROM HIT

       DO 20 J=1,PLAT
       IF(SUNK(J).LT.0)GO TO 30
       20 CONTINUE
       WRITE(6,200)
       200 FORMAT(' SUNK DIMENSIONING EXCEEDED-RECORDED BUT NOT FILED')

       PLACE SUNK SHIP IN SUNK MATRIX

       DO 40 J=1,ITAR
       SUNK(J)*ITAR
       PLATFM(SUNK(J),20)=0
       40 RETURN

       END
SUBROUTINE MOVE

MOVES ALL PLATFORMS TO THEIR NEW LOCATIONS
SUNK SHIPS ARE SKIPPED

IMPLICIT INTEGER (A-H,P-Z)
COMMON PLATFORM(100,37), DETECT(100,4), NEP(29,3), HIT(100,3), SUNK(100)
1, WATER(100), NAME(100,6), BUFFER(100,8), BLUPAS(6), REDPAS(6),
12, ZIPLAT, IAIR, TISUF, ISUB, ADV, REV, EV, ERATIC, BLAIR, REDAIR, BLUSUR,
13, REDSUR, BLUSUB, REDSUB, PBAIR1, PBAIR2, PRAIR1, PRAIR2, PBSUR1, PBSUR2,
14, WPASUR1, PASUR2, PBSUB1, PBSUB2, PASUB1, PASUB2
DO 10 1=1,-platform
10 IF (platform(1,201).EQ.0) GO TO 10
platform(1,3)=platform(1,5)
CONTINUE
RETURN
END
SUBROUTINE SEARCH

SEARCH IS THE MASTER SUBROUTINE FOR EVALUATING ALL SEARCH MISSIONS BY ALL PLATFORMS

IMPLICIT INTEGER (A-H,P-Z)
COMMON PLATFM(100,371), DETECT(100,4), WEP(20,3), HIT(100,3), SUNK(100)
1, WATER(100), NAME(100,6), BUFFER(100,8), BLUPAS(6), REDPAS(6),
2, PLATFI, IAIR, ISURF, ISUB, ADV, AEV, PEV, ERATIC, BLAIR, REDAIR, BLUSUR,
3, REDSUR, BLUSUB, REDSUB, PBAIR1, PBAIR2, PRAIR1, PRAIR2, PBSUR1, PBSUR2,
4, PSUR1, PSUR2, PBSUB1, PBSUB2, PASUB1, PASUB2
ISURAI = IAIR + ISURF
15 ISHIP = ISURF + ISUB

SURFACE AND AIR UNITS CONDUCTING ACTIVE MISSIONS AGAINST SUBSURFACE UNITS

IF (ISURAI .EQ. 0) GO TO 45
DO 20 I = 1, ISURAI
20 IF (PLATFM(I,7).EQ.0.AND.PLATFM(I,7).NE.2) GO TO 20
IF (PLATFM(I,20).EQ.0) GO TO 20
CALL ADVSUB(I)
DO 30 J = 1, ISUB
30 IF (PLATFM(J,7).EQ.0.AND.PLATFM(J,7).NE.2) GO TO 30
IF (PLATFM(J,20).EQ.0) GO TO 30
CALL ADVSUB(J)
CALL DETEK(I, J, I, DIST)
CONTINUE
20 CONTINUE

SURFACE AND AIR UNITS CONDUCTING A PASSIVE MISSION AGAINST SUBSURFACE UNITS

DO 40 I = 1, ISURAI
40 IF (PLATFM(I,7).EQ.0.AND.PLATFM(I,7).NE.2) GO TO 40
IF (PLATFM(I,20).EQ.0) GO TO 40
CALL ADVSUB(I)
DO 50 J = 1, ISUB
50 IF (PLATFM(J,7).EQ.0.AND.PLATFM(J,7).NE.2) GO TO 50
IF (PLATFM(J,20).EQ.0) GO TO 50
CALL ADVSUB(J)
CONTINUE
40 CONTINUE

233
CALL PEVSUB(I,JJ,DIST)

IF(PDV.LT.PEVIGO)GO TO 30

CALL DETEK(I,JJ,O,DIST)

30 CONTINUE

40 CONTINUE

C

C

SUBSURFACE UNITS CONDUCTING AN ACTIVE MISSION
AGAINST SURFACE AND SUBSURFACE UNITS

C

4S

I=1+IR+ISURF

IF(PLATFM(I,7).EQ.0.OR.PLATFM(I,7).EQ.2)GO TO 60

IF(PLATFM(I,20).EQ.0)GO TO 60

CALL ADYSUB(I)

DO 50 J=1,ISUB

J=J+IR

IF(I.EQ.JJ)GO TO 50

IF(PLATFM(JJ,20).EQ.0)GO TO 50

CALL AEVSUB(I,JJ,DIST)

IF(ADV.LT.AEV)GO TO 50

CALL DETEK(I,JJ,DIST)

50 CONTINUE

60 CONTINUE

RETURN

END

C

C

SUBSURFACE UNITS CONDUCTING A PASSIVE MISSION
AGAINST SURFACE AND SUBSURFACE UNITS

C

C

DO 80 I=1,ISUB

I=I+IR+ISURF

IF(PLATFM(I,7).EQ.1.OR.PLATFM(I,7).EQ.2)GO TO 80

IF(PLATFM(I,20).EQ.0)GO TO 80

CALL PDVSUB(I)

DO 70 J=1,ISUB

J=J+IR

IF(I.EQ.JJ)GO TO 70

IF(PDV.LT.PEV)GO TO 70

CALL PEVSUB(I,JJ,DIST)

70 CONTINUE

80 CONTINUE

RETURN
SUBROUTINE ADVSUB(I)

THIS SUBROUTINE COMPUTES THE ACTIVE DETECTION VALUE (ADV) FOR ALL PLATFORMS WITH ACTIVE SEARCH MISSIONS

IMPLICIT INTEGER (A-M,P-Z)

COMMON PLATFM(100,27), DETECT(100,5), MEP(29,3), HIT(100,3), SUNK(100),
1, WARE(100), TAME(100,6), BUFFER(100,6), BLUPAS (6), REDPAS(6),
2, PLAT, IAIR, ISURF, ISUB, ADV, PAV, PEV, ERATIC, BLUR, REDAIR, BLUSUR,
3, MEDSUR, BLUSUB, REDSUB, PBAIN, PBAIN2, PBAIN2, PBAIN2, PBSUR1, PBSUR2,
4, FPSUR1, FPSUR2, PBSUB1, PBSUB2, PBSUB2, PBSUB2

ADV+PLATFM(I,13) = ADV+PLATFM(I,13)
IF (PLATFM(I,6, .EQ.0)) GO TO 10
ADV = ADV + PLATFM(I,11)
IF (PLATFM(I,1, .GE. 1)) ADV = ADV - 1
IF (PLATFM(I,4, .GE. 3)) ADV = ADV - 1
10 IF (ERATIC .EQ. 1) ADV = ADV - 2
ADV = ADV - PLATFM(I,6)
IF (ADV .LT. 0) ADV = 0
RETURN
END
SUBROUTINE AEVSUB(I,J,DIST)

C
C THIS SUBROUTINE COMPUTES ACTIVE EVASION VALUES (REV) FOR ALL SURFACE AND
SUBSURFACE UNITS TO COMPARE WITH ACTIVE MISSION PLATFORMS' ACTIVE DETECTION
VALUE (HOV).

C
IMPLICIT INTEGER (A-H,P-Z)
REAL T
COMMON PLATFM(100,3), DETECT(100,4), WEP(29,3), HIT(100,3), SUNK(100),
WATER(100), NAME(100,61), BUFFER(100,61), REDPAS(61), REDAIR(61),
BLUPAS(61), BLUSUB(61), BLAIR(61), BLUSUR(61),
SREDSUR(61), BREDSUR(61), BPBASUR(61), PBUR(61),
BPBASUR2(61), PBU(61),
BLUSUB(100), PLATFM(J,3), DIST

T=DIST
IF (REV.EQ.0) REV=1
REV=REV+PLATFM(J,6)
RETURN
END
SUBROUTINE PDVSUB(I)

C THIS SUBROUTINE COMPUTES THE PASSIVE DETECTION
C VALUE (PDV) FOR ALL PLATFORMS THAT HAVE
C PASSIVE SEARCH MISSIONS

IMPLICIT INTEGER (A-H,P-Z)

COMMON PLATFM(100,37), DETECT(100,4), WEP(29,3), HIT(100,3), SUNK(100)
1, WATER(100), NAME(100,6), BUFFER(100,8), BLUPAS(6), REDPAS(6),
2 PLAT, I AIR, ISURF, ISUB, ADV, PDV, REV, ERATIC, BLURAIR, REDAIR, BLSUR,
3 REDSUR, BLUSUB, REDSUB, I Bair, I Bair2, PAIR1, PAIR2, PBSUR1, PBSUR2,
4 PRSUR1, PRSUR2, PBSUB1, PBSUB2, PBSUR2

PDV=PLATFM(I,14)
1 IF (PLATFM(I,8).EQ.0) GO TO 10
2 PDV=PDV+PLATFM(I,11)
3 IF (PLATFM(I,25).EQ.1) GO TO 10
4 IF (PLATFM(I,4).GE.1) PDV=PDV-1
5 IF (PLATFM(I,4).GE.31) PDV=PDV-1
6 IF (ERATIC.EQ.1) PDV=PDV-2
7 PDV=PDV-PLATFM(I,6)
8 IF (PDV.LT.0) PDV=0
9 RETURN
10 ENDD
SUBROUTINE PEVSUBII(J, DIST1)

C THIS SUBROUTINE COMPUTES PASSIVE EVASION VALUES (PEV)
C FOR ALL SURFACE AND SUBSURFACE UNITS TO COMPARE
C WITH PASSIVE MISSION PLATFORMS' PASSIVE DETECTION
C VALUE (PDV)

IMPLICIT INTEGER (A-H,P-Z)

REAL T
COMMON PLATFM(100,37), DETECT(100,4), WEP(29,3), HIT(100,3), SUNK(100)
WATER(100), NAME(100,61), BUFFER(100,81), BLUPAS(61), REDPAS(61),
PLAT.IAIR, ISURF, ISUB, ADV, PDV, REV, PEV, ERATIC, BLURIA, REDRIA, BLUSR,
PLUSR, BLUSUB, REDSUB, PBAIR1, PBAIR2, PBAIR3, PBAIR4, PBAIR5, PBAIR6,
CALL DISTANIPLATFM(I,33, PLATFM(J,33, 015T)

T=DIST
PEV=T/10.+99
IF (PEV.EQ.0) PEV=0
PEV=PEV+PLATFM(J,61)+PLATFM(J,6)
IF (PLATFM(J,7), .EQ. 1, OR, PLATFM(J,7), .EQ. 3) PEV=PEV-2
IF (PLATFM(J,1), .GT. 0) PEV=PEV-2
IF (PEV.LT.0) PEV=0
IF (PLATFM(J,4), .EQ. 0, AND, (PLATFM(J,7), .EQ. 0, OR, PLATFM(J,7), .EQ. 2),
1 AND, PLATFM(J,1), .EQ. 0) PEV=0
RETURN
END
SUBROUTINE DETEK(I,J,AP,DIST)

C THIS SUBROUTINE FILES ALL DETECTIONS
IN DETECT MATRIX (SEARCHER,TARGET)

C
IMPLICIT INTEGER(A-Z)

COMMON PLATFORM(100,37), DETECT(100,4), NEP(29,3), MIT(100,3), SUNK(100)
1, WATER(100), NAME(100,6), BUFFER(100,8), BLURAS(6), RERAS(6),
2 IPR1, IPR2, IPR3, IPR4, IPR5, IPR6, IPR7, IPR8, IPR9, IPR10,
3 PR1, PR2, PR3, PR4, PR5, PR6, PR7, PR8, PR9, PR10,
4 SP1, SP2, SP3, SP4, SP5, SP6, SP7, SP8, SP9, SP10,
5 USA1, USA2, USA3, USA4, USA5, USA6, USA7, USA8, USA9, USA10,
6 USA21, USA22, USA23, USA24, USA25, USA26, USA27, USA28, USA29, USA30,
7 USA31, USA32, USA33, USA34, USA35, USA36, USA37, USA38, USA39, USA40,
8 USA41, USA42, USA43, USA44, USA45, USA46, USA47, USA48, USA49, USA50,
9 USA51, USA52, USA53, USA54, USA55, USA56, USA57, USA58, USA59, USA60,
10 USA61, USA62, USA63, USA64, USA65, USA66, USA67, USA68, USA69, USA70,
11 USA71, USA72, USA73, USA74, USA75, USA76, USA77, USA78, USA79, USA80,
12 USA81, USA82, USA83, USA84, USA85, USA86, USA87, USA88, USA89, USA90,
13 USA91, USA92, USA93, USA94, USA95, USA96, USA97, USA98, USA99, USA100,
14 USA101, USA102, USA103, USA104, USA105, USA106, USA107, USA108, USA109, USA110,
15 USA111, USA112, USA113, USA114, USA115, USA116, USA117, USA118, USA119, USA120,
16 USA121, USA122, USA123, USA124, USA125, USA126, USA127, USA128, USA129, USA130,
17 USA131, USA132, USA133, USA134, USA135, USA136, USA137, USA138, USA139, USA140,
18 USA141, USA142, USA143, USA144, USA145, USA146, USA147, USA148, USA149, USA150,
19 USA151, USA152, USA153, USA154, USA155, USA156, USA157, USA158, USA159, USA160,
20 USA161, USA162, USA163, USA164, USA165, USA166, USA167, USA168, USA169, USA170,
21 USA171, USA172, USA173, USA174, USA175, USA176, USA177, USA178, USA179, USA180,
22 USA181, USA182, USA183, USA184, USA185, USA186, USA187, USA188, USA189, USA190,
23 USA191, USA192, USA193, USA194, USA195, USA196, USA197, USA198, USA199, USA200,
24 USA201, USA202, USA203, USA204, USA205, USA206, USA207, USA208, USA209, USA210,
25 USA211, USA212, USA213, USA214, USA215, USA216, USA217, USA218, USA219, USA220,
26 USA221, USA222, USA223, USA224, USA225, USA226, USA227, USA228, USA229, USA230,
27 USA231, USA232, USA233, USA234, USA235, USA236, USA237, USA238, USA239, USA240,
28 USA241, USA242, USA243, USA244, USA245, USA246, USA247, USA248, USA249, USA250,
29 USA251, USA252, USA253, USA254, USA255, USA256, USA257, USA258, USA259, USA260,
30 USA261, USA262, USA263, USA264, USA265, USA266, USA267, USA268, USA269, USA270,
31 USA271, USA272, USA273, USA274, USA275, USA276, USA277, USA278, USA279, USA280,
32 USA281, USA282, USA283, USA284, USA285, USA286, USA287, USA288, USA289, USA290,
33 USA291, USA292, USA293, USA294, USA295, USA296, USA297, USA298, USA299, USA300,
34 USA301, USA302, USA303, USA304, USA305, USA306, USA307, USA308, USA309, USA310,
35 USA311, USA312, USA313, USA314, USA315, USA316, USA317, USA318, USA319, USA320,
36 USA321, USA322, USA323, USA324, USA325, USA326, USA327, USA328, USA329, USA330,
SUBROUTINE NONVRT (INPUT, OUTPUT)
C
C THIS SUBROUTINE WILL TAKE A LIST OF NUMBERS IN CHARACTER FORMAT
C AND RETURN AN INTEGER NUMBER. IT WILL DISREGARD ANY LEADING
C BLANKS.
C
IMPLICIT INTEGER (A-Z)
DIMENSION NUMBER(101), X(701), INPUT(70)
DATA NUMBER/'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'/
DATA BLANK/*''*/
DATA DOT/'.'/
DIMENSION INPUT(70)
OUTPUT = 0
DO 1 K = 1, 70
   IF (INPUT(K).EQ.BLANK) GO TO 1
   IFNUM = K
1    CONTINUE
10   IFNUM = IFNUM + 1
20   DO 6 J = 1, IFNUM
      IF (INPUT(J).NE.BLANK.AND.INPUT(J).NE.DOT) GO TO 3
      X(J) = 0
25      CONTINUE
26   X(J) = J - 1
27   CONTINUE
30  CONTINUE
35   CONTINUE
40   CONTINUE
45   INDEX = INUM
50   DO 7 J = 1, INDEX
55      OUTPUT = OUTPUT + (X(INDEX) = (10***(J-1))
60      CONTINUE
70   INDEX = INDEX - 1
75   CONTINUE
80   RETURN
END
SUBROUTINE INVAT1(INPUT, IPUT)

C
C THIS SUBROUTINE INSPECTS A LINE OF CHARACTERS AND RETURNS THE
C FIRST ONE DISREGARDING ANY LEADING BLANKS.
C
IMPLICIT INTEGER (A-Z)
DIMENSION INPUT(70)
DATA BLANK/*/ */
DO 1 I=1, 70
1 IF (INPUT(I).EQ.BLANK) GO TO 1
INPUT=INPUT(I)
GO TO 2
1 CONTINUE
2 RETURN
END

SUBROUTINE INVAT2(INPUT, IPUT)

C
C THIS SUBROUTINE INSPECTS A LINE OF CHARACTERS AND RETURNS THE
C FIRST SIX CHARACTERS DISREGARDING ANY LEADING BLANKS.
C
IMPLICIT INTEGER (A-Z)
DIMENSION INPUT(70), IPUT(6)
DATA BLANK/*/ */
DO 1 I=1, 70
1 IF (INPUT(I).EQ.BLANK) GO TO 1
L=1
DO 2 J=1, 6
IPUT(J)=INPUT(L+J)
2 CONTINUE
GO TO 3
1 CONTINUE
3 RETURN
SUBROUTINE SCCLASS(INAME, INDEX, NAMESH)
C THIS SUBROUTINE COMPARES NAME(I) TO THE NAMES IN THE
4 C NAMESH AND RETURNS THE ROW INDEX OF THE NAME THAT MATCHES.
5 C IF THERE IS NO MATCH, INDEX IS RETURNED = 0.
6 C
7 IMPLICIT INTEGER (A-Z)
8 DIMENSION NAMESH(130,81), NAME(8)
9 INDEX=0
10 DO 1 I=1,100
11 IF(INAME(I).NE.NAMESH(I,1)) GO TO 1
12 IF(INAME(I).NE.NAMESH(I,2)) GO TO 1
13 IF(INAME(I).NE.NAMESH(I,3)) GO TO 1
14 IF(INAME(I).NE.NAMESH(I,4)) GO TO 1
15 INDEX=1
16 GO TO 2
17 1 CONTINUE
18 2 RETURN
19 END

SUBROUTINE NAMETIIINPUT(INDEX, NAME, AC1, AC2, SUR1, SUR2, SUB1, SUB2)
C
C THIS SUBROUTINE COMPARES INPUT(I) TO THE NAMES IN THE ARRAY NAME
C AND RETURNS THE ROW INDEX OF THE NAME THAT MATCHES.
C IF THERE IS NO MATCH, INDEX IS RETURNED = 0.
C
IMPLICIT INTEGER (A-Z)
DIMENSION INPUT(100,6), NAME(100,6)
INDEX=0
DO 10 I=1,100
IF((I.GE.AC1.AND.I.LE.AC2).OR.(I.GE.SUR1.AND.I.LE.SUR2).OR.
(I.GE.SUB1.AND.I.LE.SUB2)) GO TO 10
10 GO TO 1
11 IF(INPUT(I).NE.NAME(I,1)) GO TO 1
12 IF(INPUT(I).NE.NAME(I,2)) GO TO 1
13 IF(INPUT(I).NE.NAME(I,3)) GO TO 1
14 IF(INPUT(I).NE.NAME(I,4)) GO TO 1
15 IF(INPUT(I).NE.NAME(I,5)) GO TO 1
16 IF(INPUT(I).NE.NAME(I,6)) GO TO 1
17 INDEX=1
18 GO TO 2
19 1 CONTINUE
20 2 RETURN
21 END
SUBROUTINE DISTAN(XX, YY, DI)

C DETE MINES HE X DISTANCE BETWEEN TWO COORDINATES
C AND RETURNS THAT DISTANCE

INTEGER (A-Z)
DIME: 510+X(2), Y(2)
ZERO = 0
ONE =
COL =
ROW =.
XX = X(1) / 100
YY = Y(1) / 100

D = 0
100 IF XX(COL) = XX(ROW) GO TO 500
200 IF XX(COL) = 200, 300, 400
210 IF XX(COL) = XX(ROW) = XX(ROW) + 1
220 XX(COL) = XX(ROW) + 1
230 GO TO 500
240 D = D + Y(COL) - Y(COL)
250 GO TO 500
260 IF XX(COL) = XX(ROW) = XX(ROW) - 1
270 XX(COL) = XX(ROW) - 1
280 GO TO 500
290 D = D - Y(COL) - Y(COL)
300 GO TO 600
310 IF XX(ROW) = XX(ROW) = 10, 20, 30
320 IF XX(ROW) = XX(ROW) + 1
330 XX(COL) = XX(ROW) + 1
340 GO TO 500
350 D = D + X(COL) - X(COL)
360 GO TO 600
370 IF XX(COL) = XX(ROW) = XX(ROW) - 1
380 XX(COL) = XX(ROW) - 1
390 GO TO 500
400 IF XX(ROW) = XX(ROW) = XX(ROW) GO TO 100
410 IF XX(COL) = XX(COL) = XX(ROW) GO TO 100
420 RETURN
430 END
APPENDIX G

PROGRAM ACCESS

The Up Scope fortran source code is residing on tape in the computer center tape library at the Naval Postgraduate School. The name of the tape is UPSCOP, as is the filename.

The following program will generate the Up Scope fortran source code in card image format (2358 cards).

1) //SNAKEPIT JOB (1445, 0195, RY91, ,, 3000), 'GRIGGS', TIME=2
   This card is the standard NPS job card with two exceptions: ",,3000" to override the punched card limit and "TIME=2" for more than 20 seconds of CPU time.
2) // EXEC PGM=IEBPPTPCH
3) // SYSPRINT DD SYSOUT=A
4) // SYSUT1 DD UNIT=3400-3, VOL=SER=UPSCOP, DISP=(OLD, KEEP),
5) // LABEL=(1, SL, ,, IN), DSN=UPSCOP
6) // SYSUT2 DD SYSOUT=B
7) // SYSIN DD *
8) PUNCH
9) /*
BIBLIOGRAPHY


INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Cameron Station
   Alexandria, Virginia 22314
   No. Copies 2

2. Library, Code 0142
   Naval Postgraduate School
   Monterey, California 93940
   No. Copies 2

3. Department Chairman, Code 55
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93940
   No. Copies 1

4. Lt Ellen Roland, Code 55Ro
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93940
   No. Copies 1

5. LCDR William C. Griggs
    Supply Department
    USS America CV-66
    FPO New York, N.Y. 09531
    No. Copies 1

6. Capt John A. Thompson
    4715 Britton Dr.
    Stillwater, OK 74135
    No. Copies 1

7. Professor R. G. Sherwin, Code 56Sg
    Department of National Security Affairs
    Naval Postgraduate School
    Monterey, California 93940
    No. Copies 1

8. Capt Bill Caveny, USAF
    AFIT/CIRD
    Wright-Patterson AFB, OH 45433
    No. Copies 1