A MODEL FOR THE EVALUATION AND IMPROVEMENT OF A COASTAL DEFENSE SYSTEM

by

Grant William Soderstrom

Project and Report submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Systems Engineering

APPROVED:

P. M. Ghare, Chairman

B. S. Blanchard

D. R. Drew

April, 1991
Blacksburg, Virginia

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Section I: Introduction

A. Analysis of Coastal Defense Systems

1. Overview

Over the last ten years, several factors have combined to make coastal defense systems increasingly important to most nations. The primary factor has been the advent of the 200 nautical mile (NM) Exclusive Economic Zone (EEZ). This gives nations the right to enforce fishing regulations far from their coasts. The second major change has been the worldwide emphasis on stopping the drug trade. Other significant factors include enforcing the ban on ivory trade as well as stopping refugee "boat people" and other illegal immigration.

The 1982 Law of the Sea convention recognized the concept of an EEZ and a nations right to claim a 200 NM EEZ limit. While an EEZ is not a nation's sovereign territory, economic encroachment (such as fishing and oil exploration) can be regulated and enforced. Prior to the 1982 Law of the Sea convention, nations had little legal recourse when their fishing banks were plundered beyond the previously recognized 12 mile limit.[1]

While fixed installations, such as offshore oil platforms, are easily regulated and licenced, most countries have been unable to enforce their fishing laws. Most third world navies and coast guards can not even detect the illegal fishing boats off their shores. As a result, large commercial fishing fleets, such as those of the Soviet Union, have seriously depleted the fish populations off the coasts of many third world nations.[2]

While the rising third world debt has made the protection of fishing resources more important than ever, this debt has rendered most nations incapable of buying a complex, sophisticated coast guard. Nevertheless, most nations have some patrol boats and therefore have some capability to stop smugglers and illegal fishermen. The addition of a few, inexpensive maritime patrol aircraft or radars could make these patrol boats more effective. Moreover, the money saved from lost revenues could pay for the added expense. However, buying the wrong equipment would be another costly mistake.[3]

2. Coastal Defense Systems

In simplest terms, a coastal defense system is a radar coverage net with patrol craft for enforcement. The radar coverage can be provided by a patrol craft's own radar or a combination of airborne or shore based radars. While other sensors also are used, radar has the longest range and continues to represent the primary sensor for coastal defense systems. Figures 1 and 2 illustrate the basic intercept scenario.

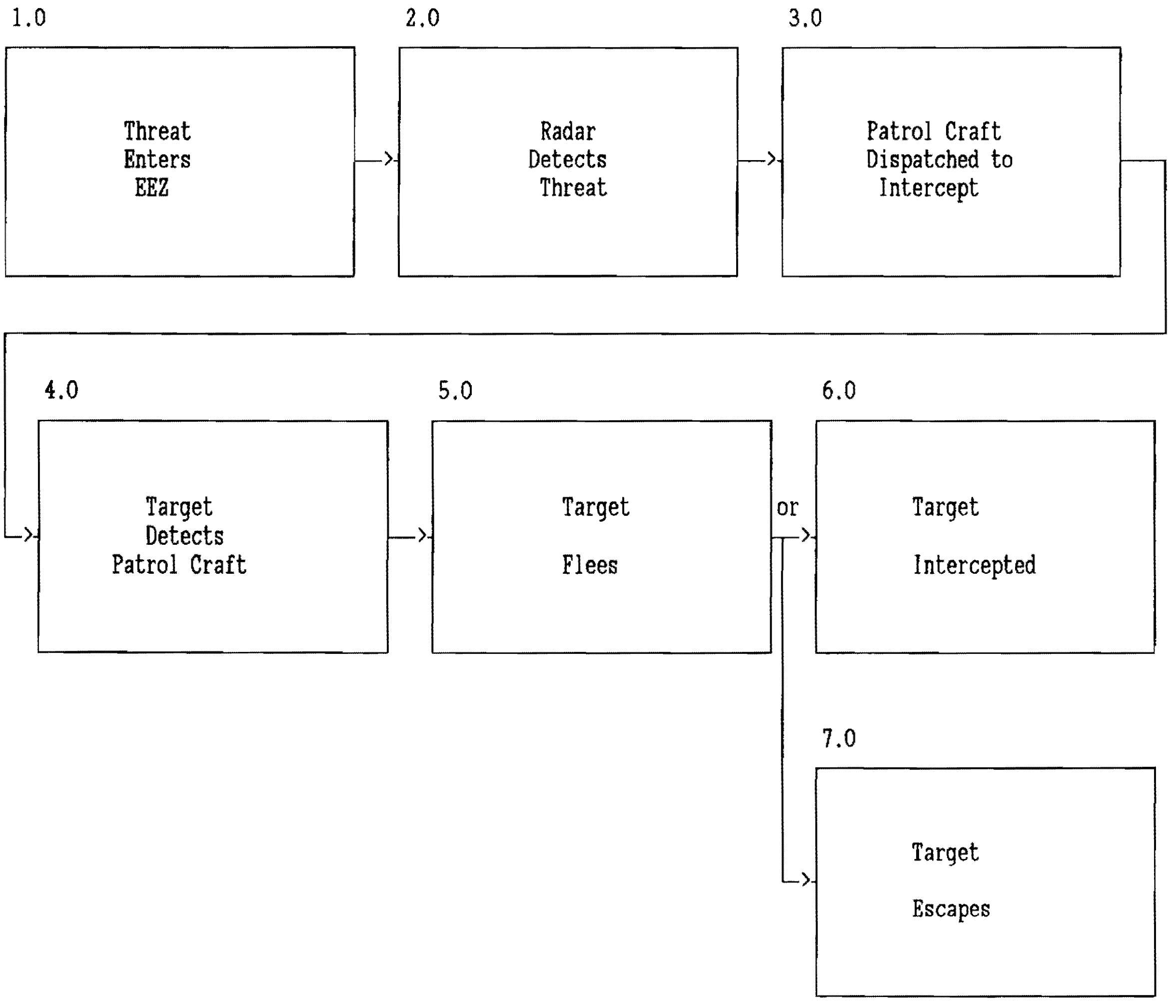


Figure 1. Threat Intercept Sequence (First Level)

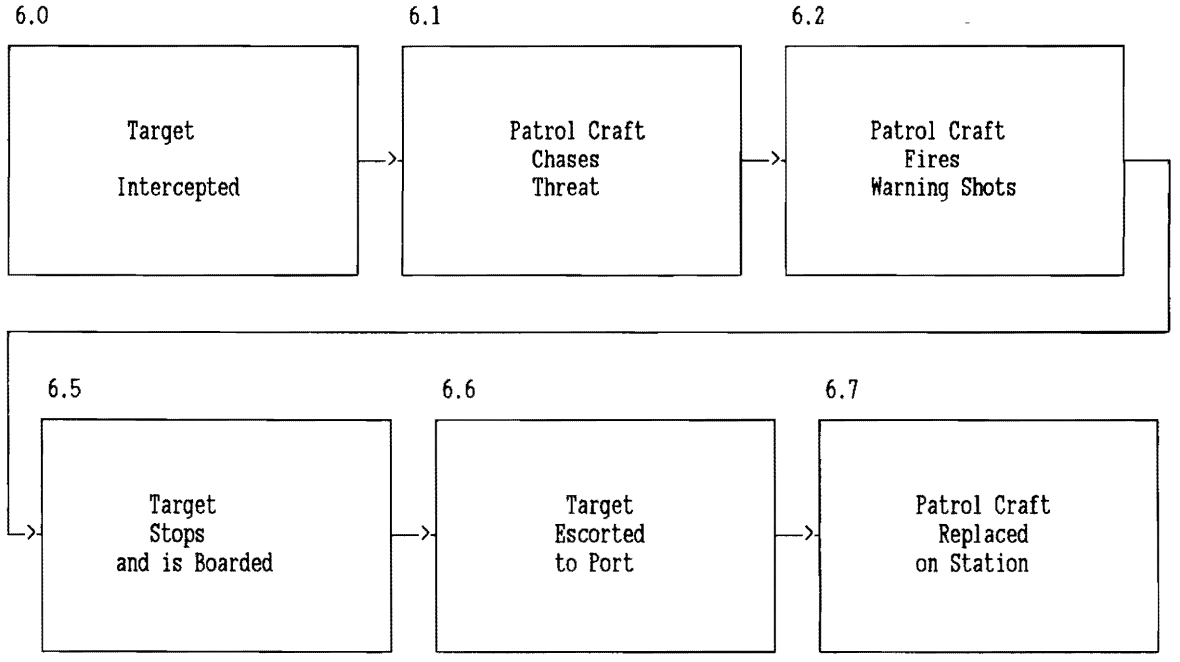


Figure 2. Threat Intercept Sequence (Second Level)

The next section is a qualitative discussion of the merits of each of the three basic elements: patrol craft, maritime patrol aircraft, and shore based radars.

a. Patrol craft

Patrol craft play the key link in any coastal defense system. While radars can detect violators, the only way to stop a transgressor on the high seas is with a patrol boat. Additionally, a suspected violator's ship often must be boarded and searched to determine if a violation has occurred.

A patrol craft can fill a variety of roles. In addition to peacetime coastal defense, they can be used to defend the coast in wartime, as well as conduct search and rescue. Because of these many demands, patrol craft are equipped with a variety of capabilities and armaments. Only those merits which pertain to peacetime coastal defense will be discussed here.

(1) A patrol craft needs a powerful radar. Nevertheless, shipborne radars have limitations. The most restrictive limitation is the mast height. Since radars are effective only to the horizon, and the horizon depends on the height of the transmitter by the following formula: [4]

Range (NM) = 1.144   Radar height above water (feet)

A radar with a theoretical range farther than the horizon is wasted capability. For this reason, shore based radars and radars on maritime patrol aircraft provide far greater coverage than those on patrol craft.

(2) A patrol craft should have seakeeping qualities (the ability to withstand heavy seas) at least as great as that of the fishermen and smugglers it expects to encounter. However, fishing is generally impossible in a sea state greater than 5 and most smugglers would be deterred by a sea state 4 or 5. Therefore, a patrol craft capable of operating in sea state 5 probably is adequate.[5]

(3) A patrol craft should have at least one large gun or mortar. Guns which make a "bang and a splash" are far more effective at making a patrol boat stop than rapid fire machine gun bursts. Additionally, a craft with larger bore guns can stop a violator with a shot across the bow from a distance.[6]

(4) A patrol craft should be faster than its targets. This ensures that the patrol craft can catch the target if it flees. However, if the patrol craft can approach within weapons range of the target undetected, the offending ship will have no time to flee. Additionally, fishing boats with nets extended are often unable to flee unless they cut their nets. Even then, 90 percent

of trawlers have maximum speeds less than 13 knots.[7] Thus, catching fishing boats after they are alerted to the patrol craft's presence is generally not a problem. Nevertheless, since fishing boats often violate remote areas of the EEZ, finding the fishing boats and traveling the long distance from shore to reach them can tax the patrol craft resources.

(5) Patrol craft have no use for missiles. Missiles are useless in stopping smugglers or illegal fishermen. They are not a credible threat for three reasons. First, few countries can afford to launch their expensive missiles (even for tests.) Second, the international outcry against an indiscriminate use of force would outweigh any gains. Finally, surface to surface missile homing systems are designed to find large metallic ships and probably could not find small wooden or fiberglass boats.[8]

b. Maritime patrol aircraft

Maritime patrol aircraft offer the most flexible method of detecting and identifying illegal fishermen and smugglers. While radar is the primary detection sensor, identification of the threat often relies on visual contact. An aircraft's speed makes it ideal for this task. Additionally, sensors such as Forward Looking Infa-Red (FLIR) enable identification at night and in foggy weather.[9]

(1) Maritime patrol aircraft should have a radar. However, this radar need not be sophisticated. An aircraft has the inherent advantage of height and speed which enables a broad area of coverage without a high powered radar. Additionally, the anti-jamming features of modern military radars are unnecessary against smugglers and illegal fishermen. [10]

(2) Maritime patrol aircraft should have good visibility. Visual identification is a key method of discriminating worthwhile targets. Additionally, visual detection remains an important backup to radar detection.

(3) Maritime patrol aircraft should have long endurance and high reliability. The length of each patrol is important to minimize the waste of transit time. The number of hours a plane can fly each year will help determine the required size of the patrol aircraft fleet.

c. Shore based radars

The principal advantage of shore based radars are their low operating and maintenance costs. Shore based radars can be manned by as few as one person (a full time crew of four in rotating shifts.) Additionally, the radar can be operated for a long time with relatively short and inexpensive maintenance periods. Nevertheless, a shore based radar cannot supplant a patrol craft or maritime patrol aircraft since it can only detect, not identify or intercept a contact.

(1) A coastal radar should have a long range. However, ranges in excess of 50 NM require far more expensive technology.

(2) A coastal radar should be simple and reliable. Since the reason to choose a coastal radar is its low operating and maintenance cost, it makes little sense to buy an unreliable one, even if it has longer range. Similarly, anti-jamming features of modern military radars are an expensive luxury that will never be used against smugglers or illegal fishermen.

(3) A coastal radar should be mobile. Mobility makes it more difficult for a smuggler to exploit the blind spots in a radar network.

B. Overview of the Simulation Model

This model is intended to be an analytical tool to help in the decision making about coastal defense system procurement options. The model will simulate any coastline, coastal force, and threat profile. The model then will yield the annual equivalent costs and savings. By running the model for each different procurement option, the options can be compared on the basis of annual cost and savings. The most economical option then can be chosen.

A POWERBASIC (compiled BASIC) program develops the Measure of Intercept Effectiveness (MIE) described below. Then the program calculates the life cycle costs and benefits.

The model begins with general information such as the geometry of coastline and the size of the Exclusive Economic Zone (EEZ). Additionally, the user inputs costs and operational parameters such as patrol craft number, speed, radar range, endurance, effectiveness, acquisition cost, operation and maintenance cost, economic life span, and home base. Similar data is entered for maritime patrol aircraft.

The model will then yield two figures of merit: the Measure of Intercept Effectiveness represents the percentage of maritime violators that will be intercepted. The Annual Equivalent Life Cycle Cost measures the annual costs of this system taking into account interest and inflation.

The model also takes inputs and calculates benefits of the coastal defense system. I have not elaborated the methods to calculate benefit factors but have conceptually outlined the sources of these calculations and have chosen reasonable numbers for my model. The benefits include:

(1) the value of preventing illegal fishing. This is complex to calculate but could be based on uncollected licence fees or lost fishing revenues and depends on whether fishing stocks are being depleted. It also accounts for the number of illegal fishing boats per day.

(2) the value of preventing drug smuggling. This includes a fraction of the nation's lost economic productivity due to drug addiction and a fraction of the current law enforcement budget. However, this must be expressed as a fraction of the current economic impact as some of the smuggling will shift to air or overland methods and some of the production may shift to domestic sources. It also accounts for the number of illegal drug smuggling boats per day.

(3) the value of preventing non-drug smuggling. This includes lost revenues from tariffs and economic impact of illegal immigration. Must be expressed as a fraction of the current economic impact as some of the smuggling will shift to air or overland methods. It also accounts for the number of illegal non-drug smuggling boats per day.

Note: the model lumps all the inbound smugglers together and all the outbound smugglers together. Therefore drug and non-drug smuggling must be subdivided into these categories and then consolidated.

Some benefits are more difficult items to quantify, including search and rescue of civilian boats and national defense. These should probably be omitted from the calculations. Search and rescue time detracts from useful intercepts and these craft would be ill suited to defend a nation from seaborne attack. However, these values, despite their limited utility, should be mentioned as over and above the quantified values when trying to get a program approved.

Section II: Detailed Description of the Simulation Model

A. Description of the Computer Program Operation

1. Assumptions

a. Threats are safe if they pass outside the EEZ (no hot pursuit.)

b. Threats are safe if they land on shore (no police coordination.)

c. Threats are uniformly distributed along the coast.

d. Fishing boats are uniformly distributed along a line approximately the same distance from shore.

e. Targets are approaching when detected.

2. Limitations

a. The coastline must run from the top of the screen to the bottom. No islands. The water may be on the left or right.

b. Map scale: the coastal segment (including the entire EEZ) should fit entirely on one screen yet the size of one screen block should not greatly exceed the radar coverage of any unit (scales between 1:2,000,000 and 1:10,000,000 are best.)

c. The patrol zones run parallel to the coast. The model does not support other patrol patterns (such as zig-zag.)

3. Narrative

The following is a narrative description of the POWERBASIC program which forms the model. Figures 3 through 9 illustrate the following description.

a. Input Segment

Country Code (CTYCODE$): Two letter country abbreviation

MAIN MENU

1. Coastal Configuration (MUST CHOOSE A PHYSICAL LAYOUT TO START)
2. Ship Class
3. Aircraft Type
4. Shore Based Radar
5. Run simulation
6. Exit <F1>

COASTAL CONFIGURATION MENU

1. Physical Layout
2. Order of Battle
3. Deployment
4. Threats

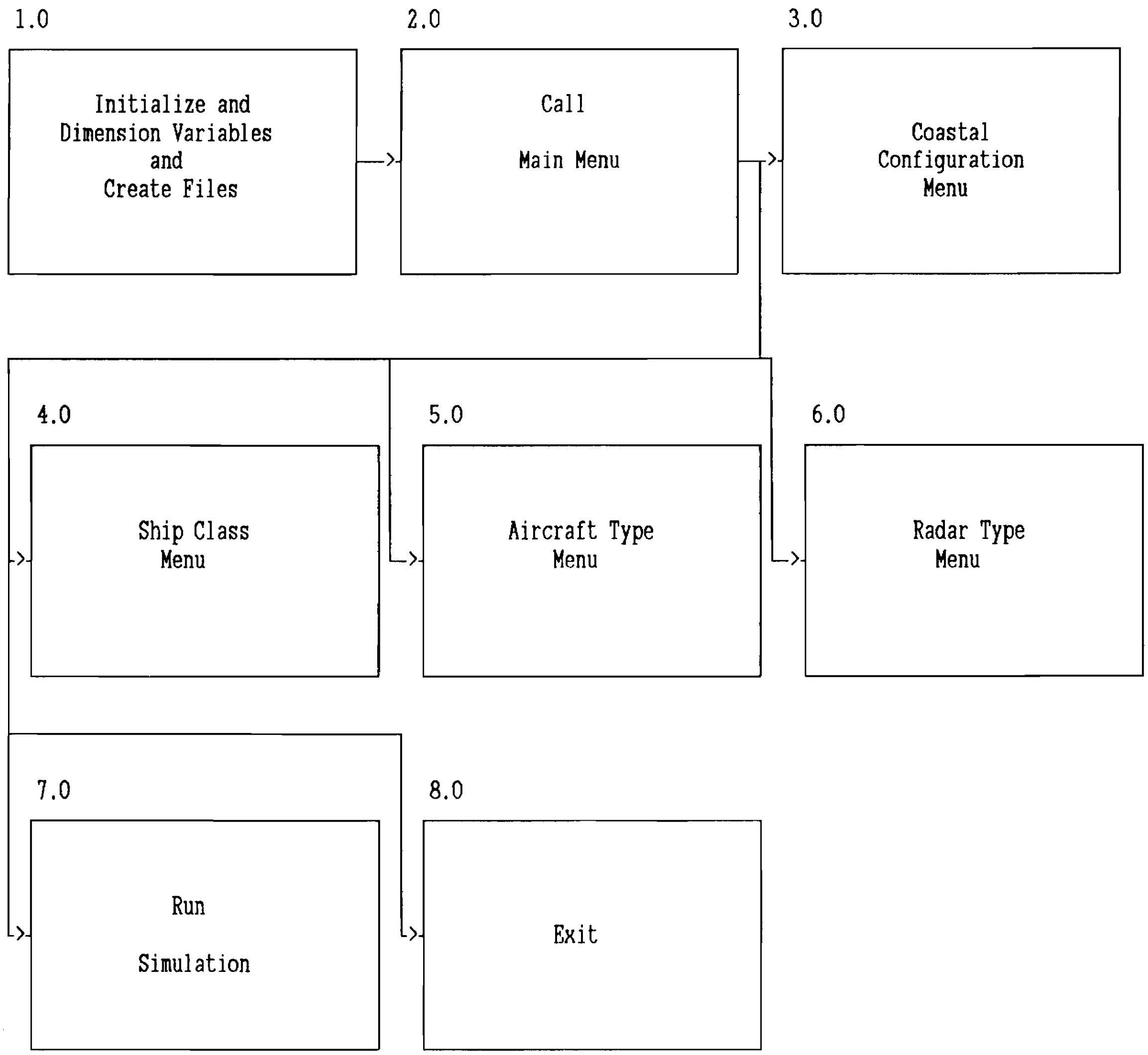


Figure 3. Computer Program Flow Diagram (First Level)

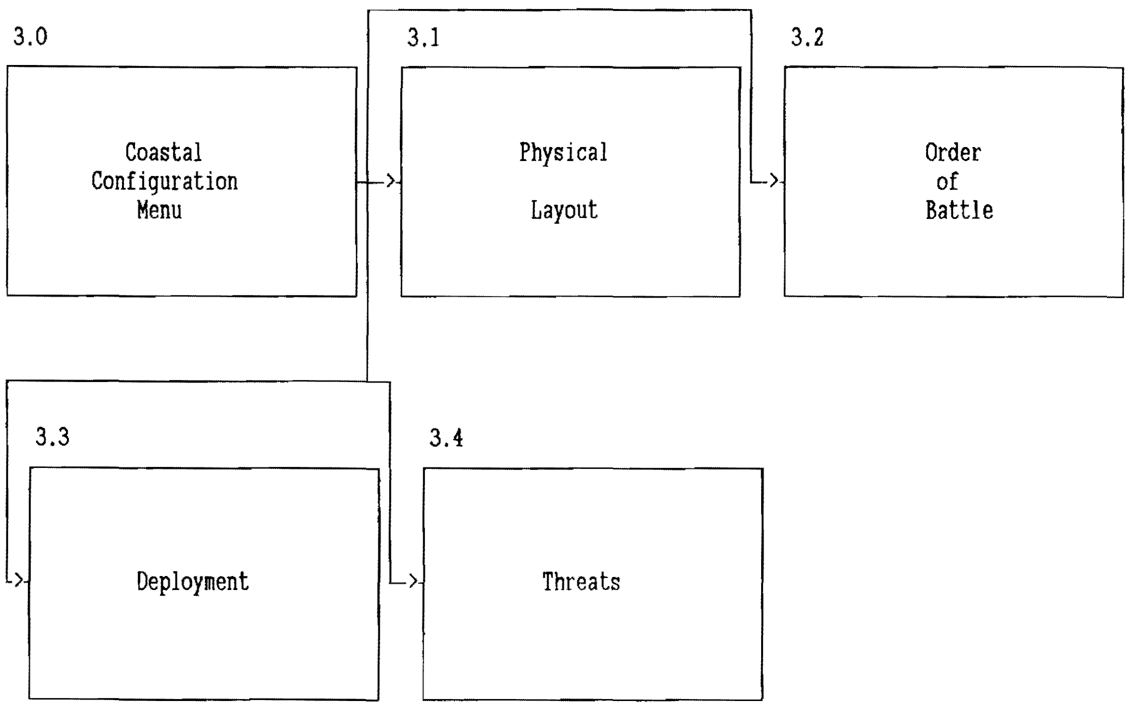


Figure 4. Coastal Configuration Menu (Second Level)

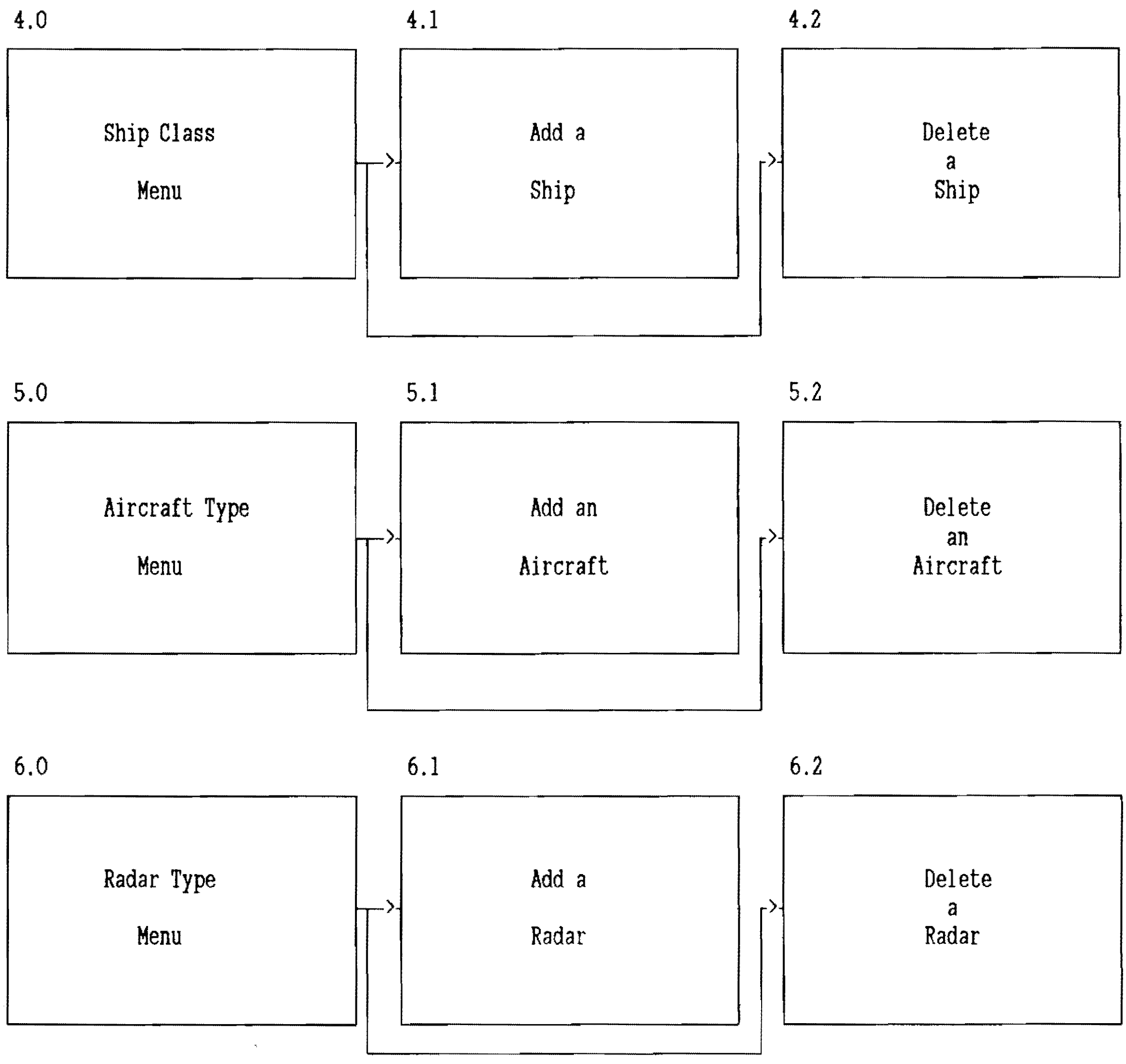


Figure 5. Equipment Files (Second Level)

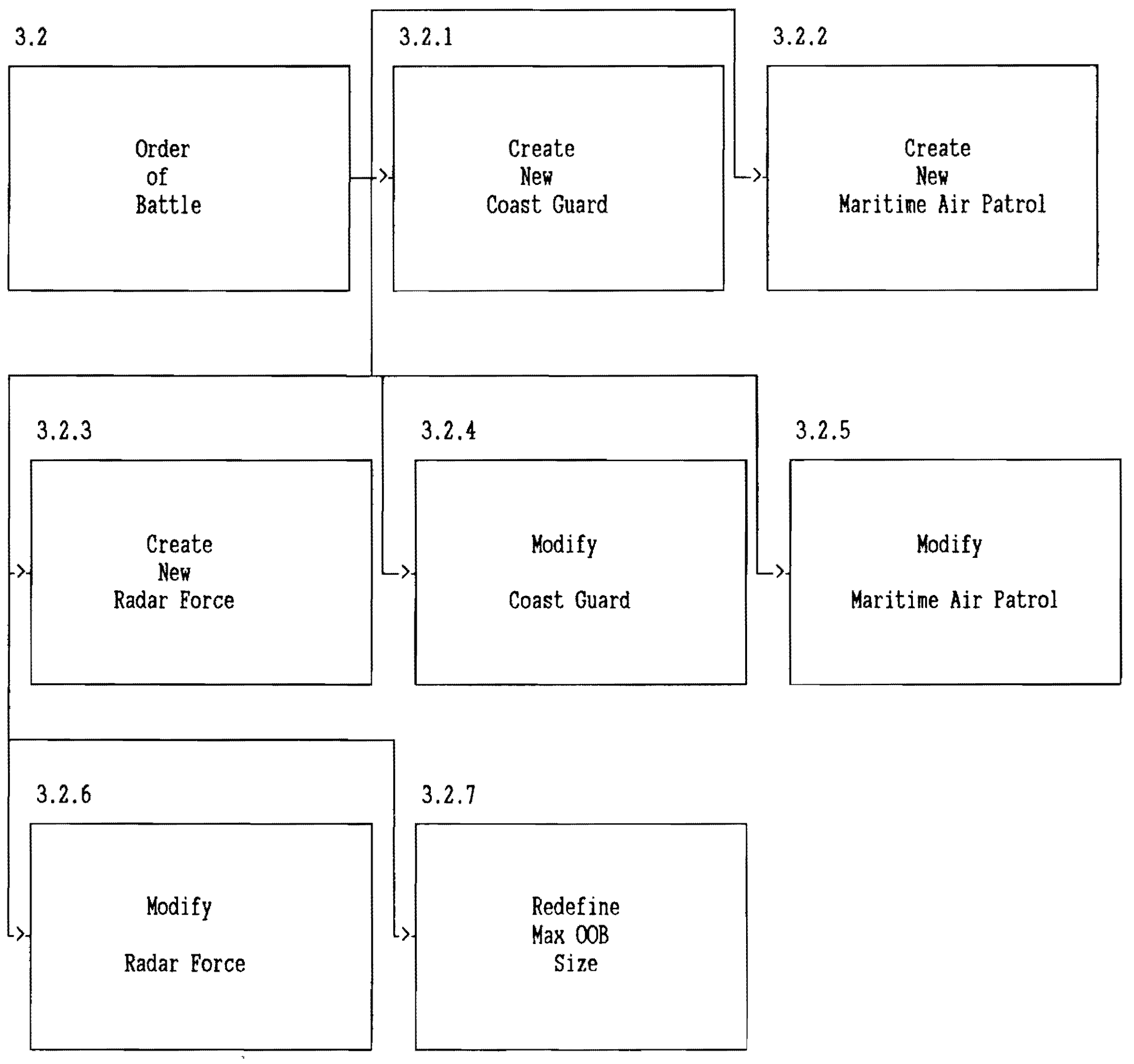


Figure 6. Order of Battle Menu (Third Level)

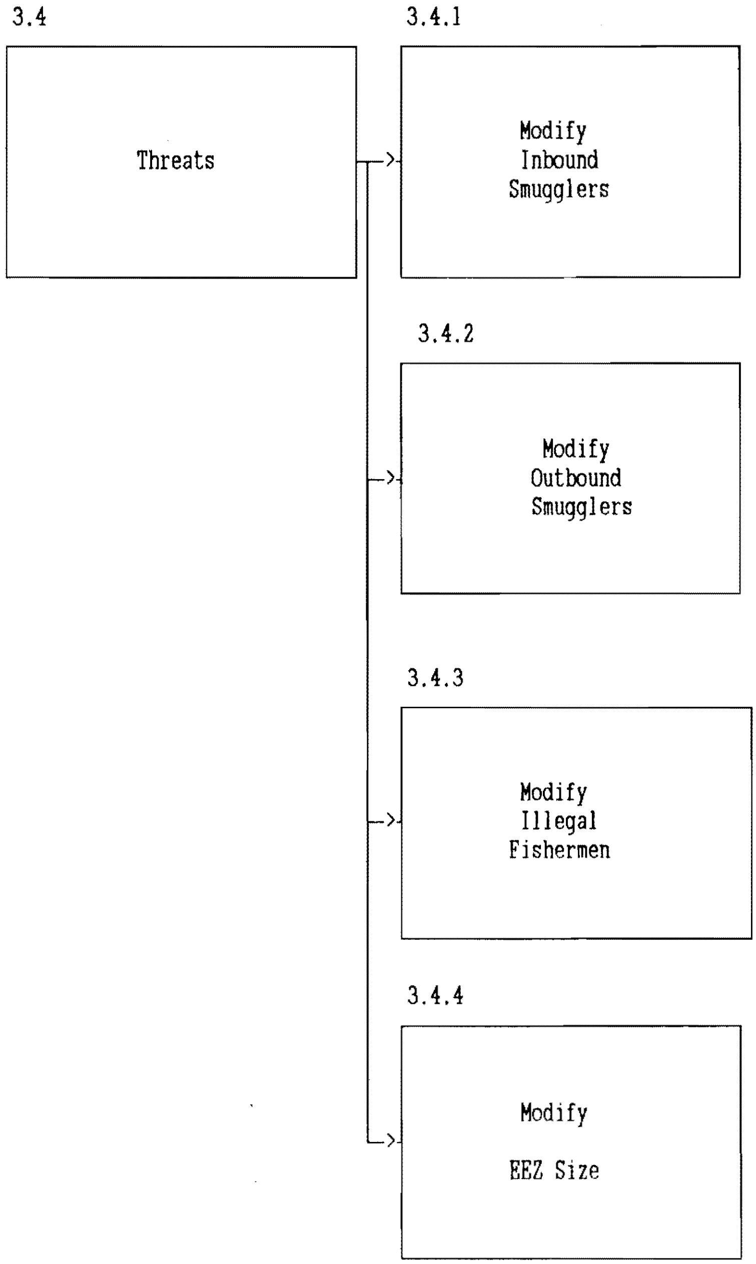


Figure 7. Threat Menu (Third Level)

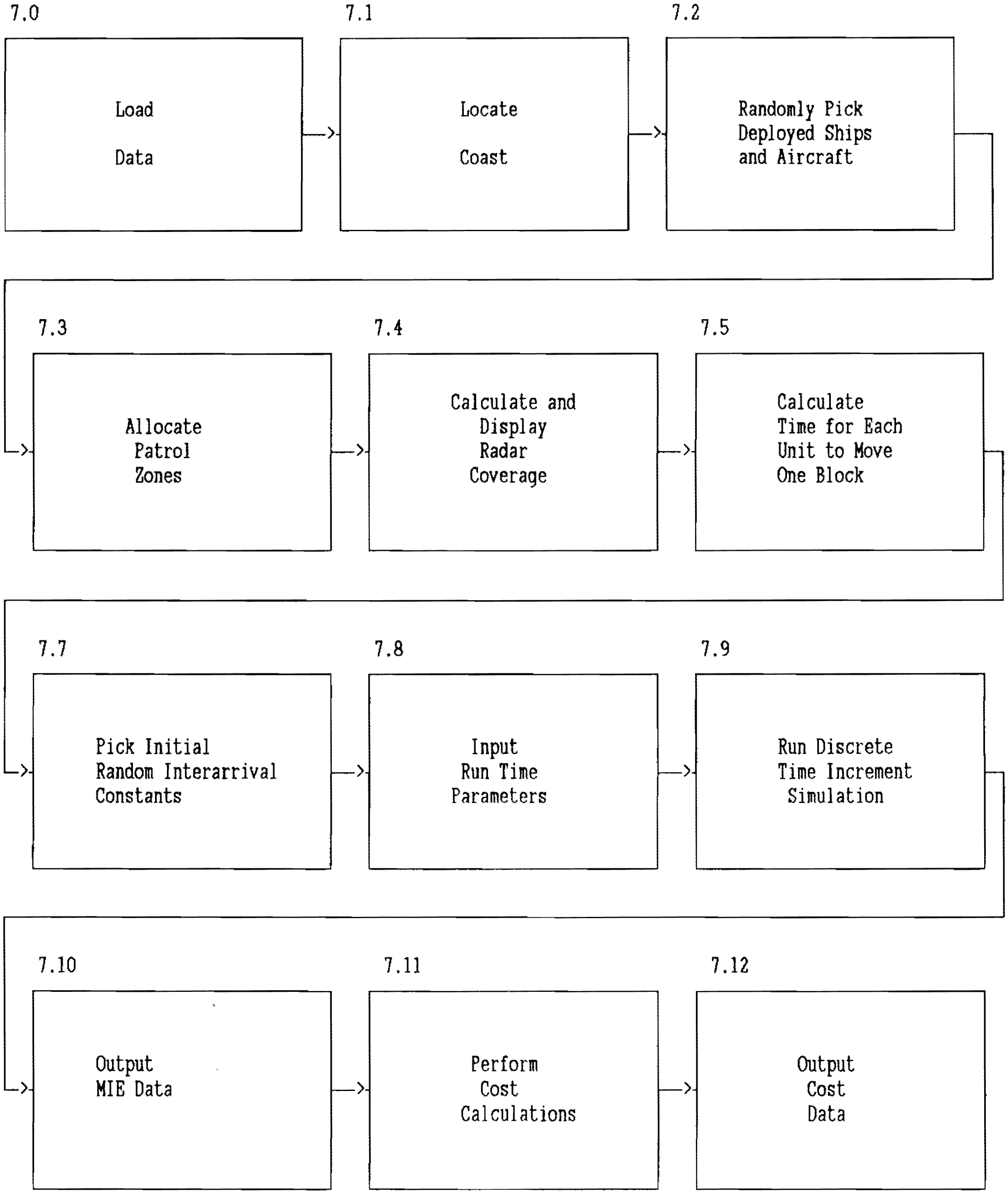


Figure 8. Simulation Routine (Second Level)

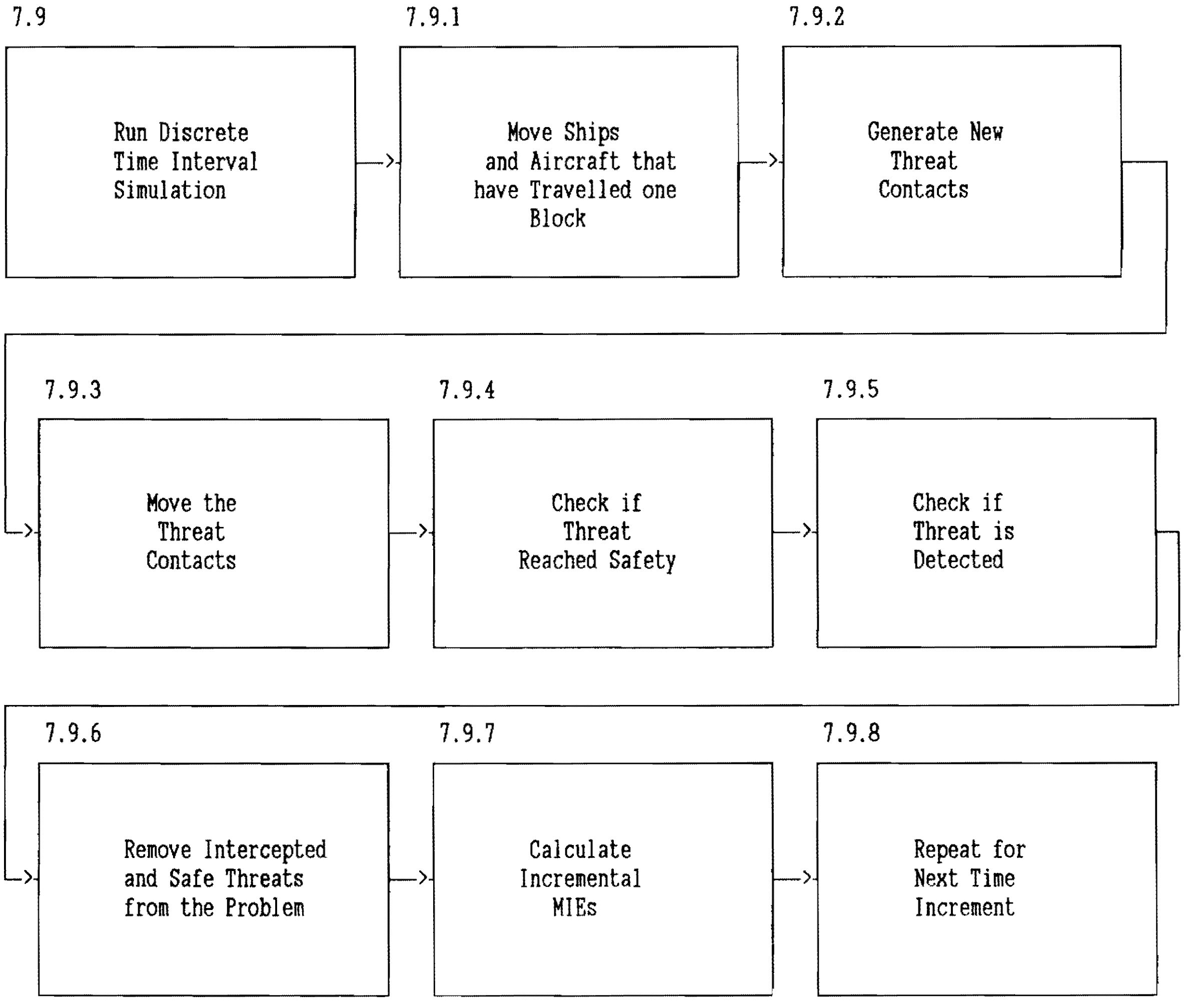


Figure 9. Simulation Routine (Third Level)

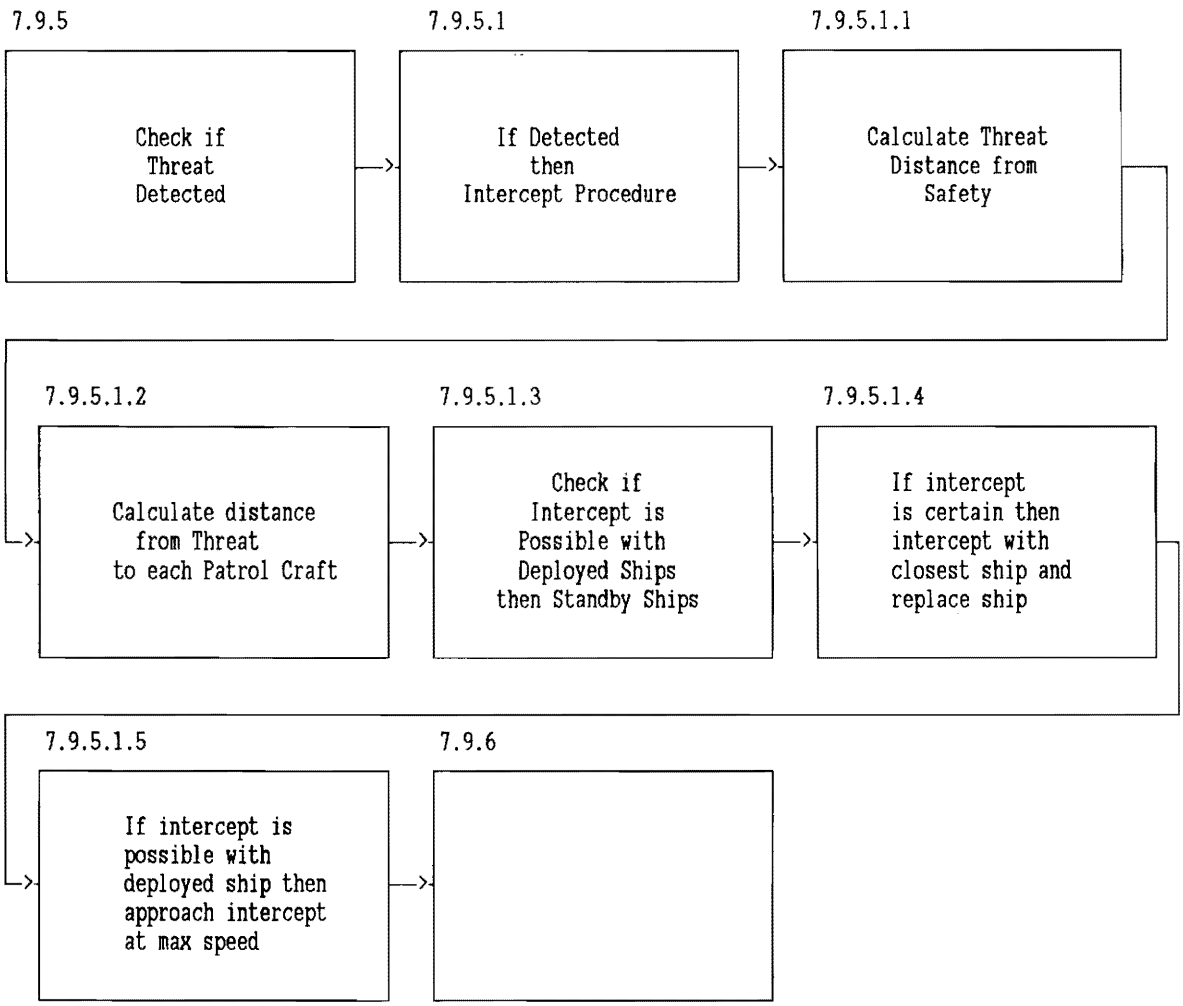


Figure 10. Intercept Procedure (Fourth and Fifth Level)

b. New Physical Layout

Location of land, water, ports, and airfields:

a) Obtain a translucent page size map of the coastal area.

b) Input the map scale (ie. 1:map scale). (SCALE!)

c) Input name of coastal area (CNAME$)

d) Tape the map up to the screen (ensure the top of the map is at the top of the screen.)

e) Fill the entire area of the screen with characters in areas covered by the map :

L: land

W: water

number the ports 1,2,3 etc.

alpha the airfields a,b,c etc.

alpha the radar stations starting with m

The program redisplays the map for you to check your work.

c. New Order of Battle

1. New Coast Guard
2. New Maritime Air Patrol
3. New Shore Based Radar Force
4. Modify Coast Guard
5. Modify Maritime Air Patrol
6. Modify Shore Based Radar Force
7. Redefine MAX Size of Each OOB

The order of battle files contain all of the variables found in the Ship Class Name, Aircraft Type, and Radar Type (See below.) Additionally, the order of battle file contains the location of each ship, plane, and radar.

New Coast Guard

Prints list of available Ship Classes

Inputs additions to the order of battle

New Maritime Air Patrol

Prints list of available aircraft types

Inputs additions to the order of battle

New Shore Based Radar Force

Prints list of available radar types

Inputs additions to the order of battle

Modify Coast Guard

Prints existing coast guard order of battle

Inputs deletions to the order of battle

Prints list of available Ship Classes

Inputs additions to the order of battle

Note: to change the location of a ship, the ship must be deleted and a new ship record must be input

Modify Maritime Air Patrol

Prints existing Maritime Air Patrol order of battle

Inputs deletions to the order of battle

Prints list of available aircraft types

Inputs additions to the order of battle

Note: to change the location of an aircraft, the aircraft must be deleted and a new record must be input

Modify Shore Based Radar Force

Prints existing Shore Based Radar Force order of battle

Inputs deletions to the order of battle

Prints list of available radar types

Inputs additions to the order of battle

Note: to change the location of an radar, the radar must be deleted and a new record must be input

Redefine MAX Size of Each OOB

The size of the random access memory on your computer will determine the maximum allowable size of your model. This statement allows you to define the number of ships, aircraft, radars, ports, and, airfields in the model.

d. New Deployment

An example of a deployment schedule is:

33% on patrol

34% on 30-minute standby

33% in maintenance

The percentage of time on patrol can be varied. Increasing the time on patrol increases the maintenance time proportionally. Additionally, maintenance and operating costs rise proportionally.

> The actual cost calculations depend on operating hours. However, the deployment determines the number of ships and aircraft deployed and therefore has a controlling effect on the operating hours and costs.

The program asks you to:

Input percent of ships deployed for ships on patrol (SPATROL!)

Input percent of aircraft deployed for aircraft on patrol (APATROL!)

The program asks you to:

Input the ship patrol distance from shore (NM): (DSPATROL!)

Input the aircraft patrol distance from shore (NM): (DAPATROL!)

e. New Threats

1. Modify Inbound Smuggling Threat File
2. Modify Outbound Smuggling Threat File
3. Modify Illegal Fishing Threat File
4. Modify EEZ size

Modify Inbound Smuggling Threat File

Prints:

the average number each day during daylight (INN!)

the average number each day during darkness (IND!)

the cost of each smuggler to the economy (INC!)

max sustained speed of smuggler craft (INS!)

the alertment distance (INA!)

Inputs new values for each

Modify Outbound Smuggling Threat File

Prints:

the average number each day during daylight (OUTN!)

the average number each day during darkness (OUTD!)

the cost of each smuggler to the economy (OUTC!)

max sustained speed of smuggler craft (OUTS!)

the alertment distance (OUTA!)

Inputs new values for each

Modify Illegal Fishing Threat File

Prints:

the average number each day during daylight (FISHN!)

the average number each day during darkness (FISHD!)

the average distance from shore (FDST!)

the cost of each fishing boat to the economy (FISHC!)

max sustained speed of fishing boat (FISHS!)

the alertment distance (FISHA!)

Inputs new values for each

Modify EEZ size

Inputs a new value for EEZ

f. MAIN MENU OPTION Ship Class

Displays a list of currently held ship classes and parameters and requires the input the following parameters for each additional ship.

SCNAM$(): Ship class name

PBSPEED!(): Max sustained speed

PBRADRNG!(): Radar range

PBENDUR: Endurance (hours at sea)

PBEFFECT: Effectiveness (1 digit numeric-one point for each 10-mm of bore diameter on the patrol boat's largest gun, a measure of intimidation when warning shots are fired.)

PBACOST!(): Acquisition cost ($)

PBOCOST!(): Operating cost ($/year)

PBMCOST!(): Maintenance cost ($/year)

PBLIFE: Life time in years

PBHOME$: Home port of patrol boat

g. MAIN MENU OPTION Aircraft Type

Displays a list of currently held aircraft types and parameters and requires the input the following parameters for each additional aircraft.

ACNAM$(): Aircraft type

ASPEED!(): Max patrol speed

ARADRNG!(): Radar range

AENDUR: Endurance (hours per patrol)

AACOST!(): Acquisition cost ($)

AOCOST!(): Operating cost ($/year)

AMCOST!(): Maintenance cost ($/year)

ALIFE: Life time in years

AHOME$: Home base of patrol aircraft

h. MAIN MENU OPTION Shore Based Radar

Displays a list of currently held radar types and parameters and requires the input the following parameters for each additional radar.

RNAM!(): Radar name

RRADRNG!(): Radar range

RACOST!(): Acquisition cost ($)

ROCOST!(): Operating cost ($/year)

RMCOST!(): Maintenance cost ($/year)

RLIFE: Life time in years

RHOME$: Home location of radar.

i. MAIN MENU OPTION Run Simulation

Simulation Routine

Calculates BLOCKSIZE. This variable tracks the size of the grid square specified in the physical layout section. It assumes that blocks on the screen are squares of 0.2 inches and uses the map scale to determine the equivalent size.

- Retrieve threat data
(IND!,INN!,INC!,INS!,INA!,OUTD!,OUTN!,OUTC!,OUTS!,OUTA!,FISHD!,FISHN!,FDST!,
FISHC!,FISHS!,FISHA!)

- Retrieve EEZ data (EEZ)

- Find locations of ports, airfields, and radars by inspecting each individual value of LAYOUT$() and selecting those that correspond to ports, airfields, and radars.
(PLAT(),PLON(),ALAT(),ALON(),RLAT(),RLON())

-Find shoreline and assign patrol areas by inspecting each pair of LAYOUT$() and selecting those that correspond to shoreline. Define which side the waters edge is on (note: shore line must extend from top to bottom of matrix, side to side will not work)
EDGE,SPLON(),APLON(),CLON(),EEZEDGE()

- Retrieve Order of Battle from files (NOBF$,AOBF$,EOBF$
(p,PBSPEED!(p),PBRADRNG!(p),PBENDUR(p),PBEFFECT(p),PBHOME(p))
(a,ASPEED!(a),ARADRNG!(a),AENDUR(a),AHOME(a))
(e,RRADRNG!(e),RHOME(e))

- Choose the initial ships and aircraft on patrol, standby, and maintenance by random number generation and comparative routine to eliminate duplication.
(SDEP(),SSTBY(),SMAINT(),ADEP(),AMAINT(),NUMSDEP,NUMSSTBY,NUMSMAINT, NUMADEP,NUMAMAINT)

- Evenly distribute ships and aircraft across coast.
(SHIPLAT(),SHIPLON(),ACFTLAT(),ACFTLON())

- Determine the time for ships, aircraft, and threats to move one block.
(TBLOCKSO,TBLOCKAO,TBLOCKINO,TBLOCKOUTO,TBLOCKFISHO)

- Establishes radar zones around each ship, aircraft, and shore based radar by calling the RADARMATRIX procedure. (Optionally prints radar coverage)

  RADARMATRIX is passed a number of variables:
  (RLAT(),RLON(),PBRADRNG!(),ARADRNG!(),RRADRNG!(),e,ID$)
  Additionally, ship and aircraft locational data and deployed, standby, and maintenance numbers are shared throughout the program and used by this routine. With this locational data and the max range of each radar, the procedure first calculates a square box

of possible radar covered areas then tests to see if each is within the actual radar coverage.

The result is the RADAR$() array which contains a "R" in every block with a radar.

- Input the days to run simulation (DAYS)

-Time increment simulation:

(0.1 hour (6 minute) intervals)

Moves ships and aircraft sequentially through each block of the patrol zones based on speed. Deletes old radar coverage. Reestablishes radar zones around each ship, and aircraft, when these move at least one block. Counts the operating hours each ship has run.

When the endurance limit of one ship or aircraft is reached, another is selected. When ships and aircraft reach their endurance limit, they are replaced by the REPLACESHIP or REPLACEACFT procedures.

REPLACESHIP: The next ship in maintenance is deployed, the next standby ship moves into maintenance, and the old deployed ship goes into standby. If a ship has insufficient operating hours to patrol for 12 hours, the next ship is taken instead. The overworked ship is returned to maintenance.

REPLACEACFT: The first maintenance aircraft is deployed and the old deployed aircraft goes into maintenance. If an aircraft has insufficient operating hours to patrol for 4 hours, the next aircraft is taken instead. The overworked aircraft is returned to maintenance.

Randomly generates smugglers and illegal fishermen based on data provided in THREATS.

-uses complementary random variables to generate an exponential distribution of arrival rate. (Poisson interarrival times). Allows one arrival of each type in each time increment.

```
      r! = RND(1)
COMPLI! = (LOG(1-r!) + LOG(r!))/2
```

This complimentary random pair is generated before the time incrementing starts and is regenerated when a threat is created.

```
TIN! = TINCHECK - ((1/INARR!)*COMPLI!)    'exponential distribution
    IF TIN! < ELT THEN
      TINCHECK = ELT
```

Similarly for outbound smugglers and illegal fishermen

Uses different threat arrival rates for day and night (INN!,IND!, to INARR!; OUTN!,OUTD! to OUTARR!; FISHN!,FISHD! to FISHARR! based on day or night)

- Keeps totals of each type of threat currently active and totals of all threats generated.
(INTHREAT,INTOT,OUTTHREAT,OUTTOT,FISHTHREAT,FISHTOT)

  - -Inbound smugglers originate at the water edge of the screen or EEZ edge (whichever is closer) and travel towards land at max speed.
  - -Outbound smugglers originate at land travel towards the water edge of the screen at max speed.
  - -Illegal fishing boats originate the distance from land specified in the threat file and travel randomly at one half max speed.

- Move threat targets similar to ships and aircraft above. If position of smuggler of illegal fisherman intersects a radar covered position (eg. RADAR$() = "R" at that location) then the intercept procedure is triggered. If threats reach safety or eliminated by intercept routine, then subroutine LESSTHREAT is triggered.

  LESSTHREAT decrements the current threat totals INTHREAT,OUTTHREAT, and FISHTHREAT and eliminates the threat contact.

Intercept procedure

- Distance and time of the threat from safety is calculated. (TSAFE!)

- Distance to the detected violator is measured to each patrol boat. (ISEPD!)

Assume that targets are approaching the patrol boat at time of detection. This is a pretty good assumption since targets going away from the patrol boat would, in general, been closer to the sensors at one time. This assumption lets us make some approximations:

1) (ISEPD! - ALERTD!) is the max distance the patrol boat must travel before alertment.

2) TALERT! = (ISEPD! - ALERTD!)/ PBSPEED!() : the time to reach the alert radius. (adds 30 minutes for ships on standby)

3) TINTERCEPT! = (ALERTD! - PBEFFECT())/ (PBSPEED!() - TSPEED!) is the time to intercept after alertment.

4) TSAFE! = SAFED/ TSPEED! is the time for the target to reach safety

IF (TSPEED! > PBSPEED!()) AND (ALERTD! > PBEFFECT()) THEN no intercept is possible.

If TSAFE > TINTERCEPT + TALERT then intercept is certain

otherwise intercept is still possible.

TPOSS!() keeps track of possibilities and chooses the closest patrol boat. If intercept is not certain, that patrol boat moves toward an intercept at max speed. Standby patrol boats to not leave port unless intercept is certain.

j. Measure of Effectiveness

Measures of Effectiveness are divided into separate categories for each of the three threats:

MIEISMUG!
MIEOSMUG!
MIEFISH!

Each is calculated by taking the number of intercepts and dividing by the number of violations. In addition, during the time increment simulation, the incremental measures of effectiveness are output to a file called TSTTST.

k. Costs and Benefits

The procedure inputs order of battle files and operating hour data, multiplies the costs by the operating hours, and totals the costs. The user must

Input interest rate (percent): (INTEREST!)
Input inflation rate (percent): (INFLATION!)

The procedure then calculates the inflation free rate:

```
IFR!= ((1+INTEREST!/100)/(1+INFLATION!/100))-1
```

Following this the procedure calculates the annual equivalent life cycle costs of acquisition for each. For example, for ships:

```
 For i = 1 to p
SALIFECOST!=PBACOST!(p)*((IFR!*(1+IFR!)^PBLIFE(p))/
(((1+IFR!)^PBLIFE(p))-1))
TOTSALIFECOST!= TOTSALIFECOST! + SALIFECOST!
 NEXT i
```

The procedure inputs threat data, the numbers of each type of intercept and multiplies to obtain savings to government. Then it converts to annual rates.

After this it calculates a net value.

Finally, the output is directed to the screen.

B. User's Manual

1. Starting the Program

The program operates best when all the files are installed on a hard disk drive. However, the program will operate (albeit slowly) from a floppy disk drive.

PowerBASIC compiles the BASIC program into an executable file. The file is called CG.EXE. Move this file (and any data files generated through previous program runs) to a hard disk drive directory. At the DOS prompt, type CG to start.

2. Exiting

Pressing <F1><Enter> at any input statement will call the exit routine. Any other time in the program, <CNTL><BREAK> will interrupt processing and exit.

3. Menus

A map of the menus is shown in figure 11.

4. Running the Program

Country Code: Enter a two letter country abbreviation

MAINMENU

1. Coastal Configuration (MUST CHOOSE A PHYSICAL LAYOUT TO START)
2. Ship Class
3. Aircraft Type
4. Shore Based Radar
5. No changes, run simulation

Select option 1

COASTAL CONFIGURATION MENU

1. Physical Layout
2. Order of Battle
3. Deployment
4. Threats

Select option 1

If the physical layout you need is already on file, then select it from the list. If not:

a) Obtain a translucent page size map of the coastal area.

b) Input name of coastal area in the format "name.country code" (eg. coast.uk)

c) Input the map scale (ie. 1:map scale).

d) Tape the map up to the screen (ensure the top of the map is at the top of the screen.)

e) Fill the entire area of the screen with characters in areas covered by the map :

L: land

W: water

number the ports 1,2,3 etc.

alpha the airfields a,b,c etc.

alpha the radar stations starting with m

The program will redisplay the map for you to check your work.

Program returns you to MAINMENU

5. Required Files

Table 1 is a list of all the files associated with this program and the menus that generate them, those with a country code as part of the name must be generated with each new country.

Table 1. Files Required

FILENAME	MENU	MENU OPTION
SCLASS	MAIN	2
ACLASS	MAIN	3
RCLASS	MAIN	4
Main Menu Option 1 to Coastal Configuration Menu		
COASTLST	COASTAL CONFIG	1
"COASTAL AREA NAME"	COASTAL CONFIG	1
CTYCODE$.DEP	COASTAL CONFIG	3
Coastal Configuration Menu Option 2 to OOB Menu		
CTYCODE$.NOB	OOB	1,4
CTYCODE$.AOB	OOB	2,5
CTYCODE$.EOB	OOB	3,6
SIZE	OOB	7
Coastal Configuration Menu Option 4 to Threat Menu		
INTHR.CTYCODE$	THREATS	1
OUTTHR.CTYCODE$	THREATS	2
FISHTHR.CTYCODE$	THREATS	3
EEZ.CTYCODE$	THREATS	4

Load each of the required files, supplying the requested information. Take care to note the units of measure requested. In general, this program deals with, nautical miles (NM), knots (KTS), and dollars per operating hour ($/OH). However, there are exceptions, for instance, radar costs are measured in dollars per year.

6. Running a Simulation

Once all the data is entered, the simulation can be run. Before the simulation starts, you will be given a opportunity to view the radar coverage. Following this you will be asked for the number of days to stabilize the model. The CG program generates a file called TSTTST which will contain the incremental measure of intercept effectiveness data. TSTTST can be imported into any graphics program or spreadsheet. A graph of TSTTST will reveal how long the model must be run to stabilize. On subsequent runs of the program, this stabilization time can be entered to exclude the bad data from the averages.

When running a simulation, you should be aware of where the model is with respect to saturation. Since a ship must spend as much time in maintenance as in operation, operating hours are limited to 12 hours per day per ship. The number of ship-hours available per day is therefore 12 times the number of ships in the fleet. Since every intercept takes approximately 4 ship-hours (two to intercept and return to port, two for the replacement ship to take station), the number of hours each day required for intercepting threats is 4 times the average number of threats. Additionally, every ship deployed takes 24 hours per day. If more ship hours are required than are available, saturation occurs. When saturation occurs, ships cannot carry out a normal deployment schedule. In this case, measures of effectiveness are significantly lower than when faced with threats less than saturation.

Numerically, saturation occurs if:

(p*12) < ((24*NUMSDEP)+(4*(IND!+INN!+OUTD!+OUTN!+FISHD!+FISHN!)))

7. Output screens

The output screens summarize the events of the simulation and the effectiveness and cost calculations. <PRINT SCREEN> if you want to retain this data. Strike any other key to advance to the next menu. Interest and inflation rates must be entered and can be varied to find the sensitivity to these parameters.

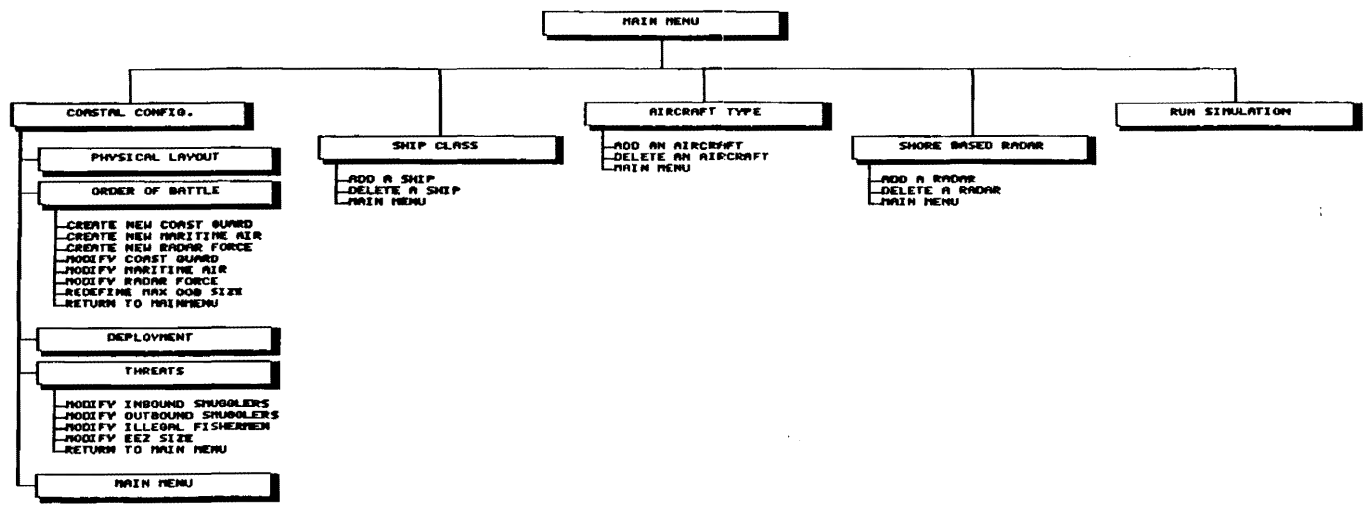


Figure 11. Menu Map

Section III: Application of the Model

1. Introduction

I have chosen Kenya as an example case because Kenya has problems with inbound smugglers (drugs), outbound smugglers (ivory), and illegal fishing. Additionally, Kenya has appealed for international assistance to solve these problems. Nevertheless, Kenya's most recent patrol craft acquisitions have been expensive missile firing patrol boats. Moreover, Kenya apparently does not use maritime patrol aircraft or shore based radars to aid in threat interdiction.

I have run several 10-day simulations of the Kenyan coastal defense system using the same threat parameters. In the base case, I have chosen the Kenyan naval order of battle prior to its most recent acquisition of two guided missile patrol craft. In subsequent cases, I have added other patrol craft, maritime patrol aircraft, and shore based radars to determine their effect on the measures of effectiveness and the costs.

Prior to its recent purchase of two Province II guided missile patrol craft, Kenya had three Vosper 32-M patrol craft, and four Brooke Marine 37-M patrol craft. [11] Table 2 shows the characteristics of these craft and other ships, planes, and radars on file in the model.

Operating and maintenance costs for these vessels have not been published. Therefore these values were estimated based on manning at $10/hour per man and fuel costs at $1.00 per gallon. Maintenance costs were estimated as double the operating costs.

The Kenyan coastline simulated in this model is shown in figure 12. The map was rotated counterclockwise and shifted to the left to ensure that the entire 200 NM EEZ fit onto the screen.

2. Length of Simulation Runs

Before starting the simulation routine, it was necessary to determine how long to run the simulation. I considered two factors. First, the simulation must be in a stable condition. For example, the generated threats must be at a quasi-equilibrium value. Figure 13 shows results of the stabilization routine which was discussed in the User's manual section. From this figure, the system appears stabile after 10 days. Second, the simulation must run long enough to get a meaningful average. Figure 14 shows that the average appears constant 10 days after the stabilization.

Table 2. Equipment Characteristics on File [11,12,13,14]

List of All Ship Classes on File

	Ship Class Name	Max Speed (KTS)	Radar Range (NM)	Max Endur. (hours)	Effect-iveness (1-9)	Acq. Cost ($)	Op Cost ($/OH)	Maint Cost ($/OH)	Life (years)
1	VOSPER 31-M	23	15	94	4	1000000	275	550	30
2	BROOKE 37-M	25	20	271	3	3000000	300	600	30
3	PROVINCE II	40	25	300	7	6E+7	450	900	30

List of All Aircraft Types on File

	Aircraft Name	Max Speed (KTS)	Radar Range (NM)	Max Endur. (hours)	Acq. Cost ($)	Op Cost ($/OH)	Maint Cost ($/OH)	Life (years)
1	DUMMY	1	0	3000	0	0	0	30
2	EMB-111	195	40	5	2400000	90	270	30
3	Do 228	180	50	4	1900000	100	300	30
4	BN-2T	140	30	6	1030000	70	210	30

List of All Radar Types on File

	Radar Name	Radar Range (NM)	Acq. Cost ($)	Op Cost ($/YR)	Maint Cost ($/YR)	Life (years)
1	DUMMY	0	0	0	0	30
2	TRS 3410	13	500000	80000	50000	30
3	PLUTO	48	4000000	80000	200000	30
4	SENTENEL	16	1000000	80000	100000	30

3. Summary of Options Explored

BASELINE CASE

Order of Battle: 3 Vosper 31-M , 4 Brooke Marine 37-M

Deployment: 10% of patrol boats deployed, 50 miles from shore

Threats: 2 Inbound smugglers, 2 outbound smugglers, and 2 fishing boats every twelve hours. Smugglers are worth $10,000 per vessel, fishermen $5,000. Smugglers travel at 15 knots, fishermen at 11 knots. Smugglers are alerted to coast guard presence at 5 miles, fishermen are constrained by their nets, hence their alertment distance is set to zero. Fishing boats average 50 miles from shore.

This baseline yielded the MIE's shown in figure 14. These MIE's were so high that a more difficult test was required to measure improvement. Therefore, I increased the number of each type of threat to 3 every twelve hours. This represents a saturated case (described in User's Guide.) This new baseline is shown in figure 15 and as the first data point in figure 16. The slow fishermen still can be caught by standby boats, but there are not enough ships in the system to keep up with the smugglers. However, figure 16 shows that simply increasing the numbers of ships (while keeping only one deployed) doesn't improve the MIEs enough to improve the net cost.

Figure 17 shows that increasing the number of ships deployed improves the MIEs and at least one combination (14 patrol craft with three deployed) has a net savings that is better than the baseline case.

Figure 18 shows that adding two maritime patrol aircraft (BN-2 Islanders) improves the net savings throughout the range of patrol boat options.

Figure 19 shows that adding three PLUTO shore based radars improves the MIEs and net savings for all patrol craft options.

The actual Kenyan Navy (with the two Province II patrol craft) yields the following results:

MIE Inbound Smugglers: .2987

MIE Outbound Smugglers: .2133

MIE Illegal Fishermen: 1

Total annual equivalent cost: $47,045,000

Net annual cost: $18,373,000

This is a much higher total and net cost than any of the other options considered. Additionally, the MIEs are worse than most other options.

4. Summary

The purchase of expensive, missile firing patrol craft probably was the worst option for the Kenyan Navy (if protecting the EEZ was their aim.) A combination of a few less expensive patrol craft, patrol aircraft, and shore based radars would do the job better and cheaper.

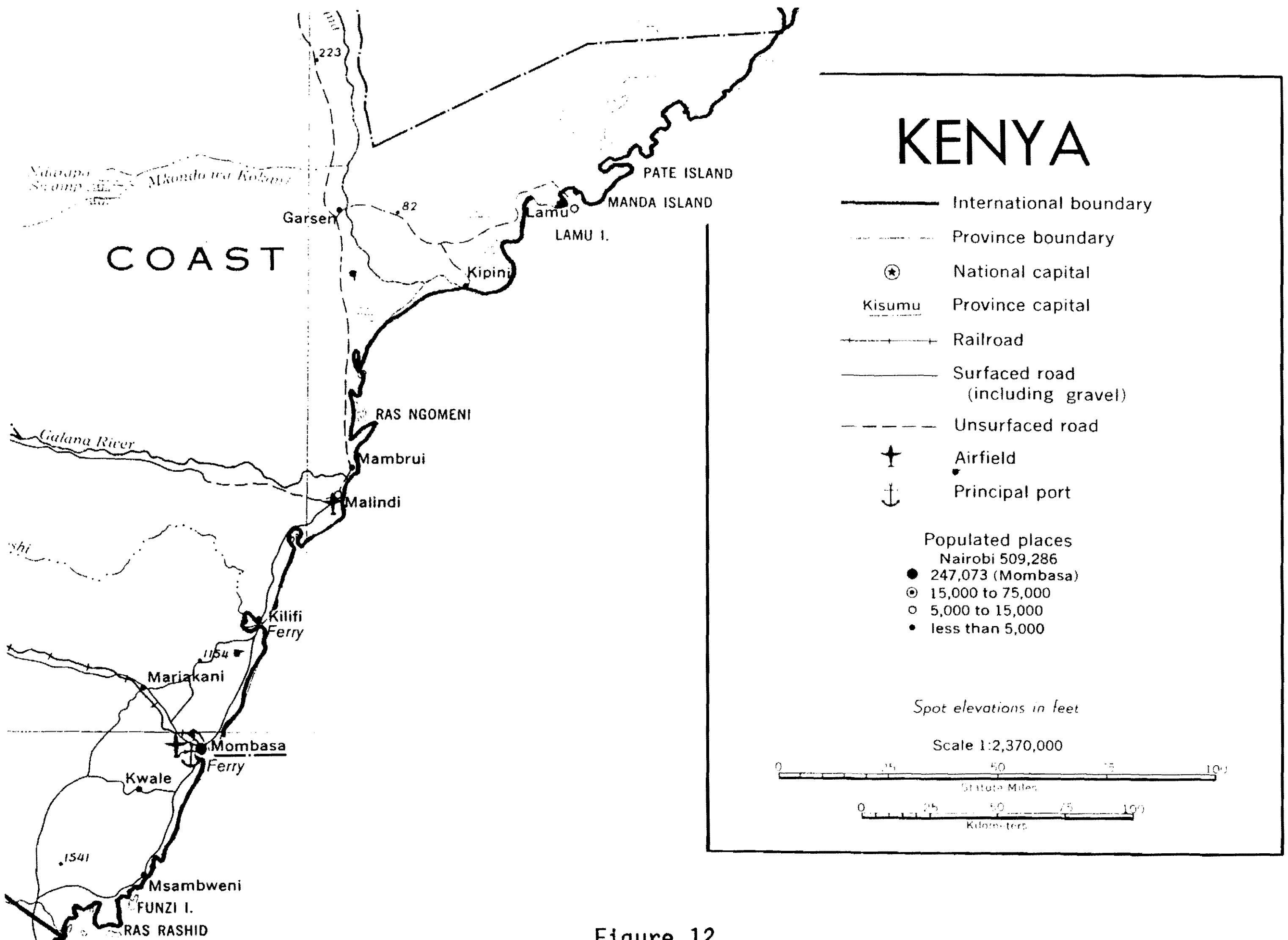

KENYA
International boundary
Province boundary
National capital
Kisumu
Province capital
Railroad
Surfaced road (including gravel)
Unsurfaced road
Airfield
Principal port
Populated places
Nairobi 509,286
247,073 (Mombasa)
15,000 to 75,000
5,000 to 15,000
less than 5,000
Spot elevations in feet
Scale 1:2,370,000
COAST
PATE ISLAND
MANDA ISLAND
LAMU I.
Lamu
Garsen
82
223
Kipini
RAS NGOMENI
Galana River
Mambrui
Malindi
Kilifi
Ferry
1154
Mariakani
Mombasa
Ferry
Kwale
1541
Msambweni
FUNZI I.
RAS RASHID

Figure 12.

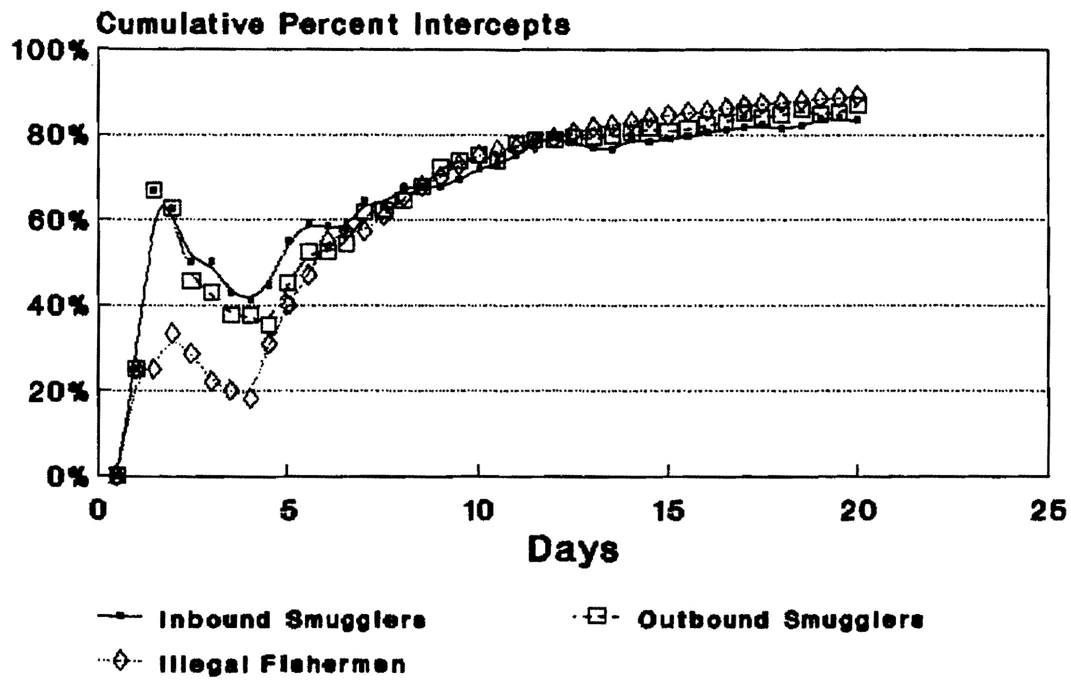


Figure 13. Stabilization Time

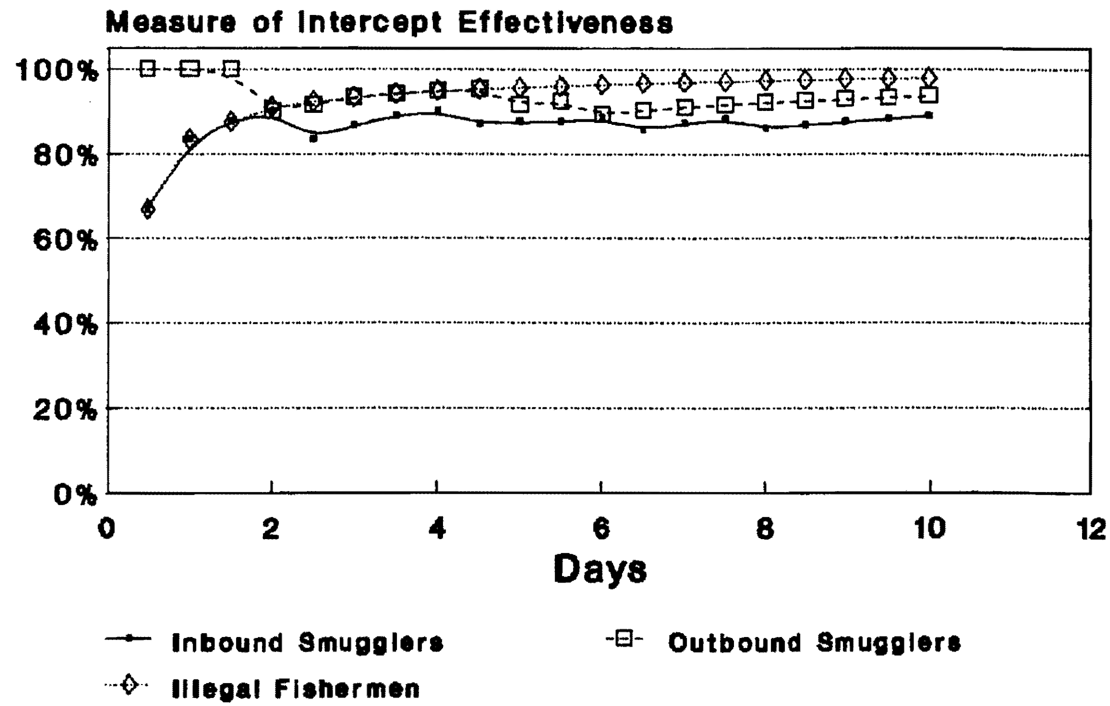


Figure 14. Baseline Case
(after 10 days stabilization)

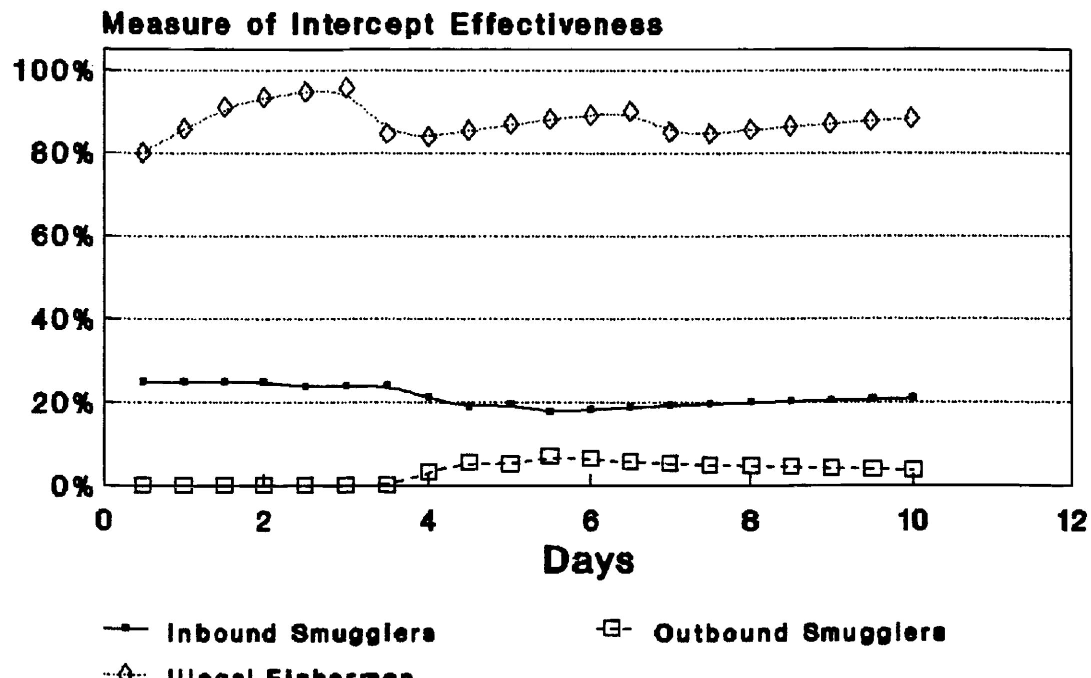
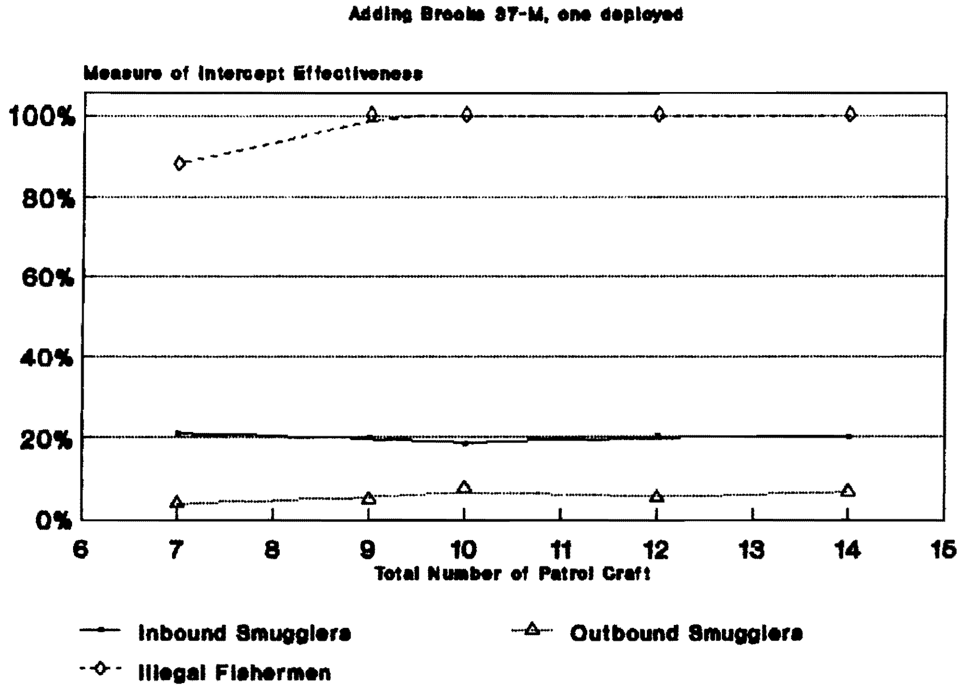


Figure 15. Baseline Case (saturated system)

(after 10 days stabilization)



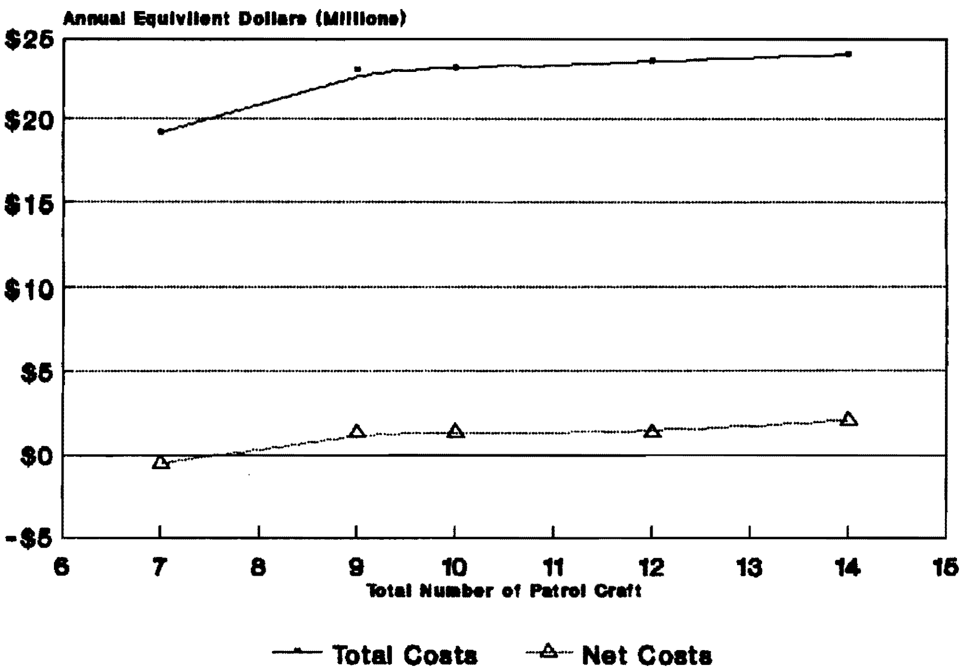


Figure 16. Effect of Adding Patrol Craft

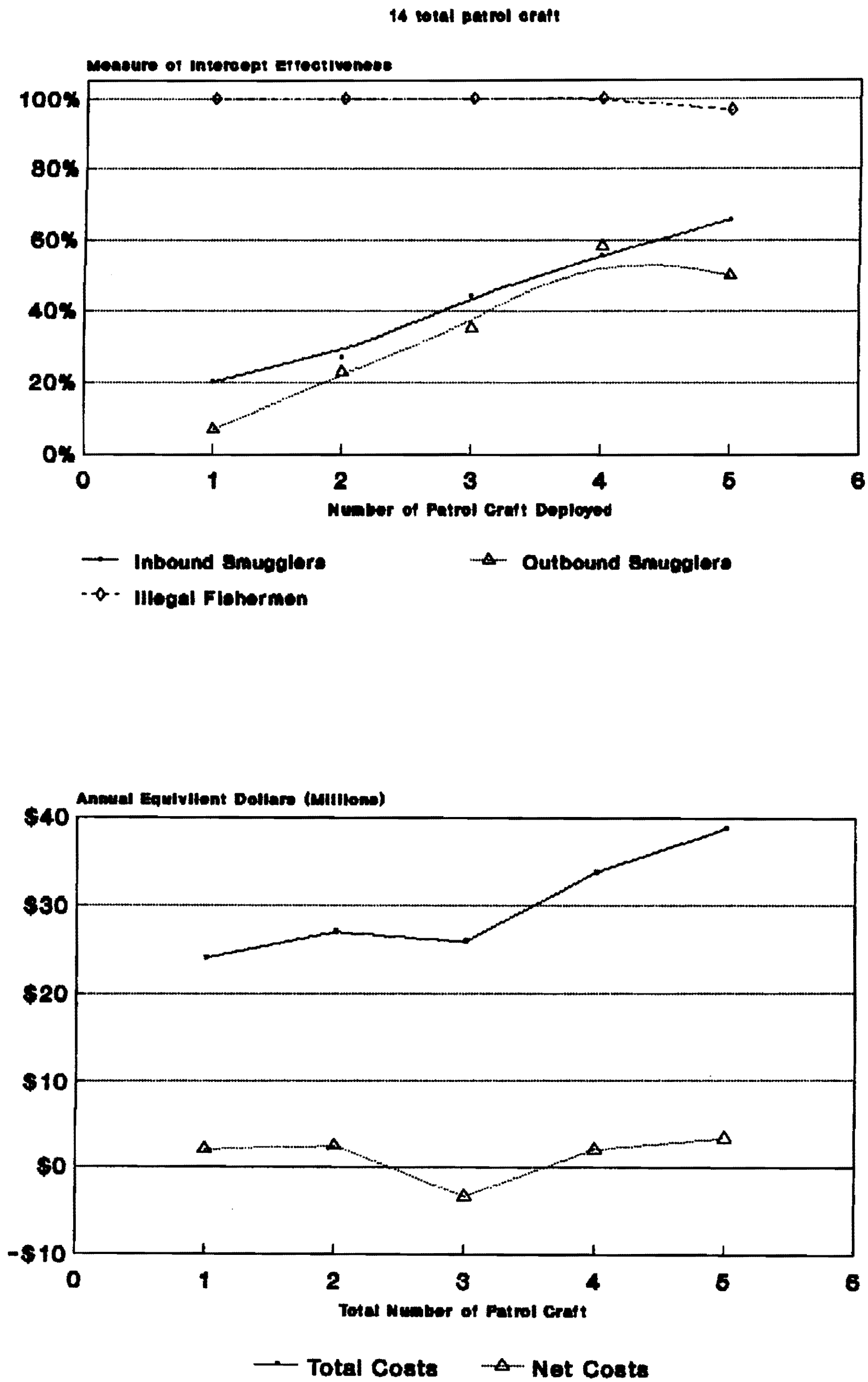
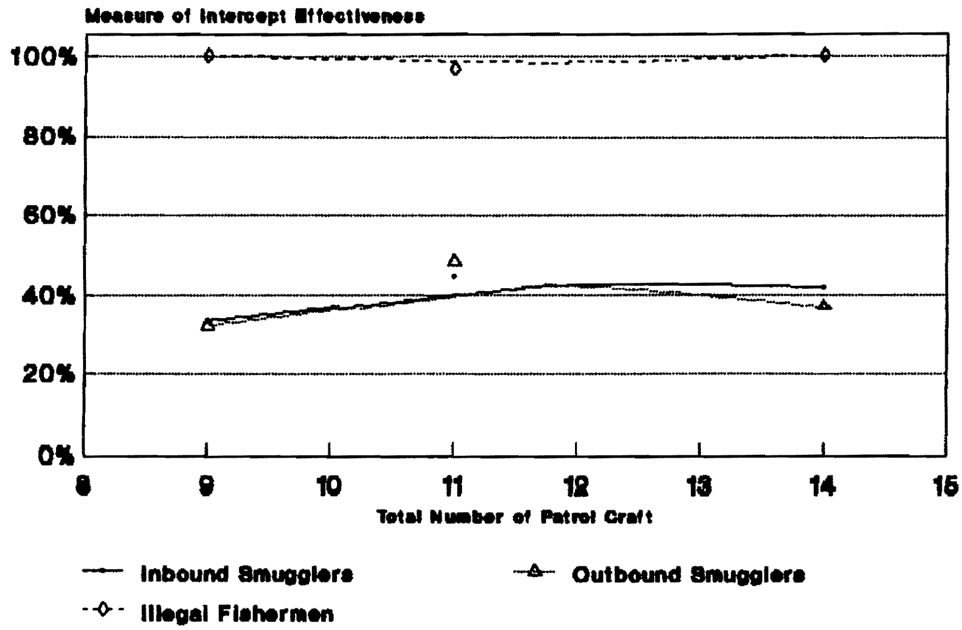


Figure 17. Effect of Increasing Deployments

two BN-2T Islander, one deployed



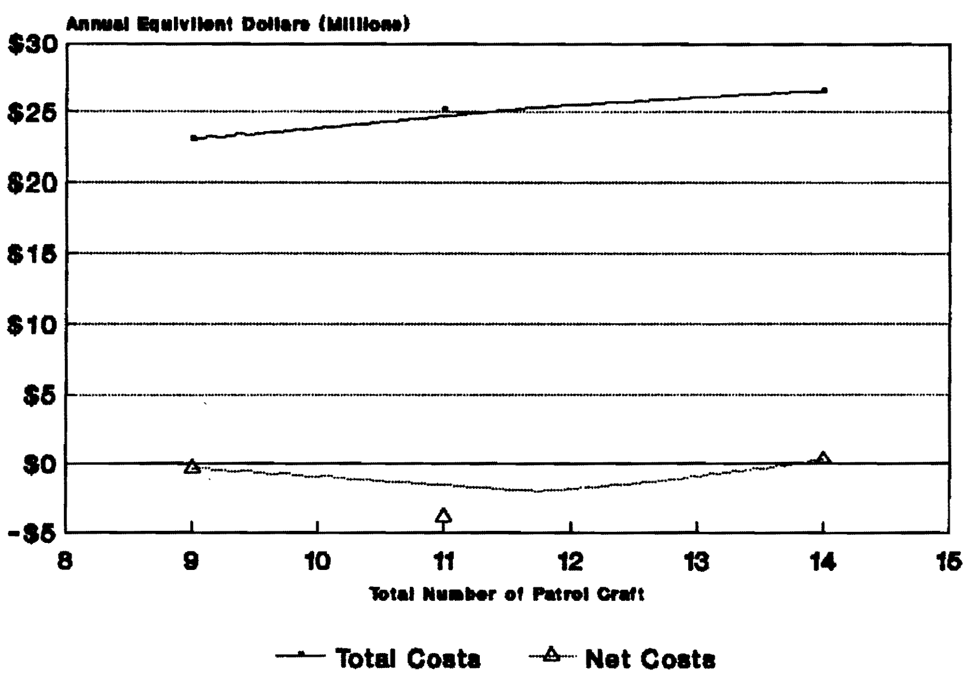


Figure 18. Effect of Adding Patrol Aircraft

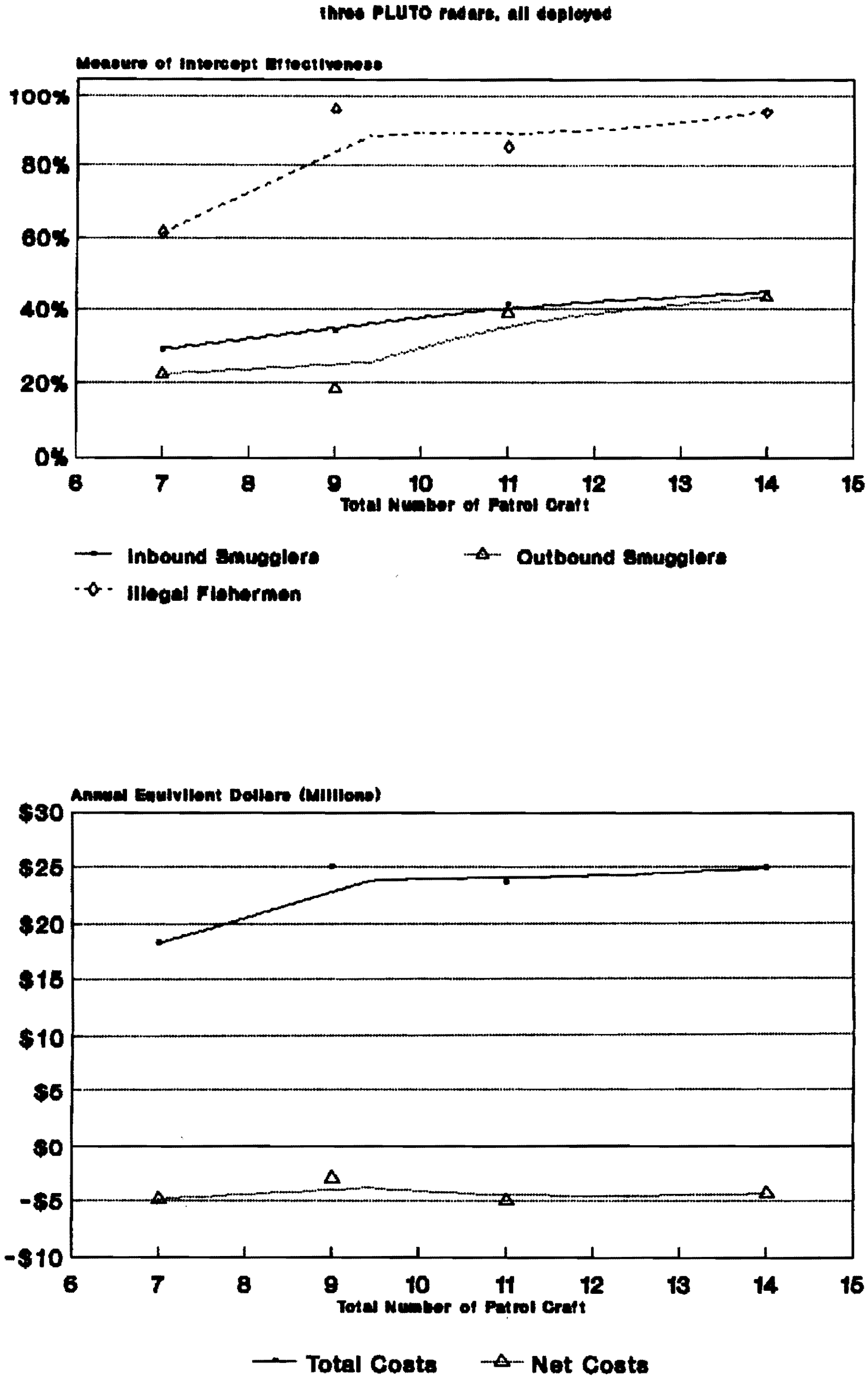


Figure 19. Effect of Adding Radars

Section IV: Conclusion

This model can be a useful tool for anyone planning an improvement of their coastal defense system. Nevertheless, an accurate evaluation will never be easy. It is difficult to obtain accurate data on the number of threats and the potential benefits of intercepting them. Once this data is obtained, the model should be run for each possible procurement alternative. Even considering only a few ship types, aircraft types, radar types, and deployment options could require thousands of simulation runs. Additionally, the alternatives should be verified against a range of threats, further multiplying the required simulation runs. However, since the potential savings is in the millions of dollars per year, this analysis would be well worth the cost and effort.

Future refinements of this model could include coordination with shore based police, so that threats not automatically considered safe when they reach shore; consideration of threats distributed non-uniformly along the coast; more complex coastal geometries such as islands or groups of islands; and more complex patrol and deployment patterns.

Section V: Summary of Literature Cited

[1] Office of the Geographer, "National Maritime Claims: 1958-1985," Geographic Research Study No. 20, (Washington DC: US Dept. of State, 1985) p. 14.

[2] James L. Woods, "US Security Assistance to Africa," African Defense, October 1989, pp. 51-55.

[3] Christopher A. Abel, "Of Sirens' Songs and Sailors," Naval Institute Proceedings, March 1991, pp. 79-82.

[4] Richard R. Hobbs, Marine Navigation 1: Piloting, (Annapolis, MD: Naval Institute Press, 1974) p. 85.

[5] David K. Brown, "A Stable, Seaworthy Platform," International Defense Review, No. 11/1990, pp. 1253-1257.

[6] Antony Preston, "Protecting the EEZ," African Defense, March 1989, pp. 51-53.

[7] Brown, p. 1255.

[8] Preston, p. 53.

[9] Eric J. Grove, "African Navies: Policing their Interests", Naval Forces, No. II/1990, pp. 59-63.

[10] Brian Walters, "Airborne Maritime Patrol", African Defence, October 1989, pp. 63-65.

[11] A. D. Baker, III, ed., Combat Fleets of the World 1984/1985, (Annapolis, MD: Naval Institute Press, 1984)

[12] Aviation Advisory Services, Limited, International Air Forces & Military Aircraft Directory, Stapleford Airfield, Essex, England, 1989.

[13] Derek Wood, Jane's World Aircraft Recognition Handbook, Jane's Publishing Company, 1985.

[14] Jane's Radar and Electronic Systems 1990-1991, Jane's Publishing Company, 1989

Appendix A: PowerBASIC Program

```
' Master's Project and Report for MS in System's Engineering
' Grant W. Soderstrom
' Spring 1991
' Written in POWERBASIC
' This program will find a Measure of Effectiveness for a Coastal
' Defense System

$stack 8000
$STRING 2

DEFINT a-z
SHARED layout$(),RADAR$(),SCALE!,ctycode$,NOBF$,AOBF$,EOBF$,DAYS
SHARED INTHR$,OUTTHR$,FISHTHR$,DEPF$,EEZ$,BLOCKSIZE!
SHARED INTOT,OUTTOT,FISHTOT,ININCPTTOT,OUTINCPTTOT,FISHINCPTTOT
SHARED PATINCPTTOT,STBYINCPTTOT,SZ,AZ,RZ,PZ,AFZ,TPT,TPT1
SHARED NUMSSTBY,NUMSMAINT,NUMSDEP,NUMADEP,NUMAMAINT,ACFTLAT(),ACFTLON()
SHARED SDEP(),SSTBY(),SMAINT(),ADEP(),AMAINT(),SHIPLAT(),SHIPLON()
SHARED SOPHOURS!(),AOPHOURS!(),TBLOCKSO!(),TBLOCKAO!(),_
       MIEISMUG!,MIEOSMUG!,MIEFISH!,SREST(),AREST()

  OPEN "SIZE" FOR INPUT AS #1      'input dimensioning variables
   INPUT #1,SZ,AZ,RZ,PZ,AFZ
  CLOSE #1

DIM layout$(30,85),RADAR$(-20:44,-20:99)
DIM SDEP(SZ),SSTBY(SZ),SMAINT(SZ),ADEP(SZ),AMAINT(AZ)
DIM SHIPLAT(SZ),SHIPLON(SZ),ACFTLAT(AZ),ACFTLON(AZ),SREST(SZ),AREST(AZ)
DIM SOPHOURS!(SZ),AOPHOURS!(AZ),TBLOCKSO!(SZ),TBLOCKAO!(AZ)

on key(1) gosub gexit
key(1) on
on error goto gerror

color 7,1,8
CLS

'Define Country, Order of Battle, and Threat Files

 INPUT "Country Code"; CTYCODE$

                                  ' Order of Battle File Names
     NOBF$ = CTYCODE$ + "." + "NOB"     ' Coast Guard
     AOBF$ = CTYCODE$ + "." + "AOB"     ' Maritime Air
     EOBF$ = CTYCODE$ + "." + "EOB"     ' RADAR

     DEPF$ = CTYCODE$ + "." + "DEP"    'Deployment data

                                         'Threat File Names
     INTHR$ = "INTHR" + "." + CTYCODE$          'Inbound Smugglers
     OUTTHR$ = "OUTTHR" + "." + CTYCODE$        'Outbound Smugglers
     FISHTHR$ = "FISHTHR" + "." + CTYCODE$      'Illegal Fisherman
```

```
     EEZ$ = "EEZ" + "." + CTYCODE$

     OPEN NOBF$ FOR APPEND AS #1     'generate necessary files
     CLOSE #1
     OPEN AOBF$ FOR APPEND AS #1
     CLOSE #1
     OPEN EOBF$ FOR APPEND AS #1
     CLOSE #1
     OPEN DEPF$ FOR APPEND AS #1
     CLOSE #1
     OPEN INTHR$ FOR APPEND AS #1
     CLOSE #1
     OPEN OUTTHR$ FOR APPEND AS #1
     CLOSE #1
     OPEN FISHTHR$ FOR APPEND AS #1
     CLOSE #1

CALL MAINMENU

SUB MAINMENU

 DO

 Print
 Print "                    MAIN MENU"
 Print
 Print
 Print "Choose one of the following (1-4 can review data or add new):"
 Print "1. Coastal Configuration (MUST LOAD PHYSICAL LAYOUT TO START)"
 Print "2. Ship Class"
 Print "3. Aircraft Type"
 Print "4. Shore Based Radar"
 Print "5. Run simulation"
 Print "6. Exit <F1>"
 Print
 Print

 100 INPUT "> ", MAINCASE

 SELECT CASE MAINCASE

   CASE = 1
     CALL COASTC
   CASE = 2
     CALL SCLASS
   CASE = 3
     CALL ATYPE
   CASE = 4
     CALL SRADAR
   CASE = 5
     CALL SIMULATE
 cls
```

```
     CALL MIE
While not instat : wend
 k$ = INKEY$
 cls
     CALL COST
While not instat : wend
 k$ = INKEY$
 cls
   CASE = 6
     GOSUB GEXIT
   CASE ELSE
     PRINT "Enter a number 1 through 5"
     GOTO 100

 END SELECT

 LOOP

END SUB


SUB COASTC


 Print
 Print "         COASTAL CONFIGURATION MENU"
 Print
 Print
 Print "Choose one of the following to review old data or input new:"
 Print "1. Physical Layout"
 Print "2. Order of Battle"
 Print "3. Deployment"
 Print "4. Threats"
 Print "5. Return to Main Menu"
 Print
 Print

200 INPUT "> ", COASTCASE

 SELECT CASE COASTCASE

   CASE = 1
     CALL PLAYOUT
   CASE = 2
     CALL OOB
   CASE = 3
     CALL DEPLOY
   CASE = 4
     CALL THREAT
   CASE = 5
     EXIT SELECT
   CASE ELSE
     PRINT "Enter a number 1 through 5"
     GOTO 200
```

```
 END SELECT

END SUB

SUB PLAYOUT  'translates a screen sized map into an array

DIM cmap$(25),cold$(20),sc!(20)

 cls
 PRINT "Coastal areas on file"
 PRINT "NAME" Tab(15) "SCALE!"
 LST$= "coastlst"
 OPEN LST$ for input as #1
    x = 0
   While not eof(1)
    x = x + 1
      Input #1, Cold$(x), SC!(x)
      Print Cold$(x) TAB(15) "1:"; SC!(x)
   wend
 close #1

 Print
 Print
 Print "Coastal area names can be up to 8 characters ";
 Print "and should end with .ctycode (eg.  east.us)"
 Print
 Input "Input name of coastal area (can be new or old) ", CNAME$

     'determines if coastal area is new and rebuilds LST$
OPEN LST$ for OUTPUT as #1
     OLD = 0
   For y = 1 to x
    if lcase$(cname$) = lcase$(cold$(y)) then
    Input "Coastal area already exists, do you want to overwrite";logi$
      if lcase$(left$(logi$,1)) <> "y"  THEN
       OLD = 1
       input "Do you want to change scale"; logi$
         if lcase$(left$(logi$,1)) = "y"  THEN
            Input "Input map scale 1:"; SC!(Y)
         END IF
      else
       Input "Input map scale 1:"; SC!(y)
      END IF
     SCALE! = SC!(Y)
    END IF
   WRITE #1, Cold$(y),SC!(y)
  Next y

 CLOSE #1

    IF OLD = 1 THEN
```

```
      GOTO DISCST
    END IF

    IF SCALE! = 0 THEN                                  'update new file
      Open LST$ for append as #1
      Input "Input map scale 1:"; SCALE!
      write #1, cname$,SCALE!
      CLOSE #1
    END If

 SCREENINPT:                               'screen input sequence

 print "when screen goes blank, fill screen with symbols
 print "(one line at a time) and then <enter> at the end of each line"
 Print "<Press any key to continue>

 While not INSTAT : wend
   k$= inkey$
 cls

 For x = 1 to 24
   LINE input, cmap$(x)
 next x


 For x= 1 to 24
   For y = 1 to 79
     layout$(x,y) = mid$ (cmap$(x),y,1)
   next y
 next x
 cls

 open cname$ for output as #3

 for x=1 to 24
   for y = 1 to 79
     write #3, layout$(x,y)
   next y
 next x

 close #3

 DISCST:

Input "Do you want to see the coast"; logi$
     lo$= lcase$(left$(logi$,1))
cls

 open cname$ for input as #4

 for x=1 to 24
   PRINT
  for y = 1 to 79
   input #4, layout$(x,y)
```

```
   If lo$ = "y" then
    bcol$ = lcase$(layout$(x,y))

    Select Case bcol$
      CASE "l"
      Color 7,6,0
      Print " ";
      CASE "w"
      color 7,1,0
      Print " ";
      CASE "a" To "i"
      color 7,6,0
      Print bcol$;
      CASE "m" to "u"
      color 2,0,0
      Print bcol$;
      CASE ELSE
      color 1,7,0
      Print bcol$;
    END SELECT
   End If
   next y
 next x

   If lo$ = "y" then

 while not instat : wend
      K$ = INKEY$

   end if

 close #4
 color 7,1,8
 cls

END SUB


SUB OOB

DIM SCNAM$(SZ),PBSPEED!(SZ),PBRADRNG!(SZ),PBENDUR(SZ),PBEFFECT(SZ),_
    PBACOST!(SZ),PBOCOST!(SZ),PBMCOST!(SZ),PBLIFE(SZ),PBHOME(SZ)
DIM ACNAM$(AZ),ASPEED!(AZ),ARADRNG!(AZ),AENDUR(AZ),_
    AACOST!(AZ),AOCOST!(AZ),AMCOST!(AZ),ALIFE(AZ),AHOME$(AZ)
DIM RNAM$(RZ),RRADRNG!(RZ),_
    RACOST!(RZ),ROCOST!(RZ),RMCOST!(RZ),RLIFE(RZ),RHOME$(RZ)


 Print
 Print "                     ORDER OF BATTLE MENU"
 Print
 Print
 Print "Do you want to:"
 Print "1. Create New Coast Guard"
```

```
 Print "2. Create New Maritime Air Patrol"
 Print "3. Create New Shore Based Radar Force"
 Print "4. Modify Coast Guard"
 Print "5. Modify Maritime Air Patrol"
 Print "6. Modify Shore Based Radar Force"
 Print "7. Redefine MAX Size of Each OOB"
 Print "8. Return to Main Menu"
 Print
 Print

400 INPUT "> ", OOBCASE

cls

 SELECT CASE OOBCASE

   CASE 1
     OPEN NOBF$ for output as #10
     CALL NOB
   CASE 2
     OPEN AOBF$ for output as #10
     CALL AOB
   CASE 3
     OPEN EOBF$ for output as #10
     CALL EOB
   CASE 4
     Print "This is the current Coast Guard Order of Battle"
     Print
     OPEN NOBF$ for input as #10
     y=1
     z=20

WHILE not EOF(10)     'print to screen one screen at a time

 Print TAB(6) "Ship" TAB(17) "Max" TAB(23) "Radar" TAB(29) "Max"_
  TAB(36) "Effect-" TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life" TAB(74) "Home"
 Print TAB(6) "Class" TAB(17) "Speed" TAB(23) "Range" TAB(29) "Endur."_
TAB(36) "iveness" TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"_
 TAB(74) "Base"
 Print TAB(6) "Name" TAB(17) "(KTS)" TAB(23) "(NM)" TAB(29) "(hours)"_
  TAB(37) "(1-9)" TAB(45) "($)" TAB(51) "($/OH)" TAB(59) "($/OH)"_
   TAB(66) "(years)"
 Print

     FOR x = y to z

  INPUT #10, SCNAM$(x),PBSPEED!(x),PBRADRNG!(x),PBENDUR(x),_
    PBEFFECT(x),PBACOST!(x),PBOCOST!(x),PBMCOST!(x),PBLIFE(x),PBHOME(x)
  print x TAB(5) SCNAM$(x) TAB(17) PBSPEED!(x) TAB(22) PBRADRNG!(x)_
    TAB(29) PBENDUR(x) TAB(37) PBEFFECT(x) TAB(41) PBACOST!(x)_
 TAB(50) PBOCOST!(x) TAB(59) PBMCOST!(x) TAB(67) PBLIFE(x)_
  TAB(75) PBHOME(x)
```

```
        If not EOF(10) then
         exit if
        else
         exit for
        End If

      NEXT x

      y = x
      z = x + 19

While not INSTAT : WEND
       K$ = INKEY$
      cls

LOOP
      CLOSE #10

     DO
      Input "Input the number of the ship to be deleted ", DN
      If DN = 0 then exit loop
      SCNAM$(DN)= ""

    OPEN NOBF$ for output as #10  'rebuild file NOBF$

     For i = 1 to x
      If SCNAM$(i) <> "" then
       Write #10, SCNAM$(i),PBSPEED!(i),PBRADRNG!(i),PBENDUR(i),_
    PBEFFECT(i),PBACOST!(i),PBOCOST!(i),PBMCOST!(i),PBLIFE(i),PBHOME(i)
      End If
     Next i
    CLOSE #10
    LOOP

      OPEN NOBF$ for append as #10
      CALL NOB

    CASE 5
      Print "This is the current Maritime Air Patrol Order of Battle"
      Print
      OPEN AOBF$ for input as #10

      y=1
      z=20

WHILE not EOF(10)     'print to screen one screen at a time

 Print TAB(6) "Aircraft" TAB(17) "Max" TAB(23) "Radar" TAB(29) "Max"_
  TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life" TAB(74) "Home"
 Print TAB(17) "Speed" TAB(23) "Range" TAB(29) "Endur."_
   TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost" TAB(74) "Base"
 Print TAB(6) "Name" TAB(17) "(KTS)" TAB(23) "(NM)" TAB(29) "(hours)"_
   TAB(45) "($)" TAB(51) "($/OH)" TAB(59) "($/OH)"_
```

```
   TAB(66) "(years)"
  Print

      FOR x = y to z

     INPUT #10, ACNAM$(x),ASPEED!(x),ARADRNG!(x),AENDUR(x),_
      AACOST!(x),AOCOST!(x),AMCOST!(x),ALIFE(x),AHOME$(x)
     print x TAB(5) ACNAM$(x) TAB(17) ASPEED!(x) TAB(22) ARADRNG!(x)_
      TAB(29) AENDUR(x) TAB(41) AACOST!(x)_
      TAB(50) AOCOST!(x) TAB(59) AMCOST!(x) TAB(67) ALIFE(x)_
       TAB(75) AHOME$(x)

        If not EOF(10) then
         exit if
        else
         exit for
        End If

      NEXT x

      y = x
      z = x + 19

 While not INSTAT : WEND
       K$ = INKEY$
      cls

 LOOP
      CLOSE #10

     DO
      Input "Input the number of the aircraft to be deleted ", DN
      If DN = 0 then exit loop
      ACNAM$(DN)= ""

    OPEN AOBF$ for output as #10  'rebuild file AOBF$

     For i = 1 to x
      If ACNAM$(i) <> "" then
        WRITE #10, ACNAM$(i),ASPEED!(i),ARADRNG!(i),AENDUR(i),_
        AACOST!(i),AOCOST!(i),AMCOST!(i),ALIFE(i),AHOME$(i)
      End If
     Next i
    CLOSE #10
     LOOP

      OPEN AOBF$ for append as #10
      CALL AOB

    CASE 6
     Print "This is the current Shore Based Radar Force Order of Battle"
      Print
      OPEN EOBF$ for input as #10
      y=1
```

```
     z=20

WHILE not EOF(10)     'print to screen one screen at a time

 Print TAB(6) "Radar" TAB(23) "Radar"_
  TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life" TAB(74) "Home"
 Print TAB(6) "" TAB(23) "Range"_
  TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost" TAB(74) "Base"
 Print TAB(6) "Name" TAB(23) "(NM)"_
  TAB(45) "($)" TAB(51) "($/YR)" TAB(59) "($/YR)"_
   TAB(66) "(years)"
 Print

     FOR x = y to z

   INPUT #10, RNAM$(x),RRADRNG!(x),_
    RACOST!(x),ROCOST!(x),RMCOST!(x),RLIFE(x),RHOME$(x)
   print x TAB(5) RNAM$(x) TAB(22) RRADRNG!(x)_
    TAB(41) RACOST!(x)_
    TAB(50) ROCOST!(x) TAB(59) RMCOST!(x) TAB(67) RLIFE(x)_
     TAB(75) RHOME$(x)

       If not EOF(10) then
        exit if
       else
        exit for
       End If
     NEXT x

     y = x
     z = x + 19

While not INSTAT : WEND
      K$ = INKEY$
     cls

LOOP

     CLOSE #10

    DO
     Input "Input the number of the radar to be deleted ", DN
     If DN = 0 then exit loop
     RNAM$(DN)= ""

   OPEN EOBF$ for output as #10  'rebuild file EOBF$

    For i = 1 to x
     If RNAM$(i) <> "" then
       Write #10, RNAM$(i),RRADRNG!(i),_
       RACOST!(i),ROCOST!(i),RMCOST!(i),RLIFE(i),RHOME$(i)
     End If
    Next i
```

```
   CLOSE #10
    LOOP

     OPEN EOBF$ for append as #10
     CALL EOB

   CASE 7

  OPEN "SIZE" FOR OUTPUT AS #1
   PRINT "Input max number of ships, aircraft, radars, ports, and "
   INPUT "airfields in model (separated by commas): ",SZ,AZ,RZ,PZ,AFZ
   WRITE #1,SZ,AZ,RZ,PZ,AFZ
  CLOSE #1

REDIM SDEP(SZ),SSTBY(SZ),SMAINT(SZ),ADEP(SZ),AMAINT(AZ),AREST(AZ)
REDIM SHIPLAT(SZ),SHIPLON(SZ),ACFTLAT(AZ),ACFTLON(AZ),SREST(SZ)
REDIM SOPHOURS!(SZ),AOPHOURS!(AZ),TBLOCKSO!(SZ),TBLOCKAO!(AZ)

   CASE 8
     EXIT SELECT
   CASE ELSE
     PRINT "Enter a number 1 through 9"
     GOTO 400

 END SELECT


END SUB


SUB NOB

DIM SCNAM$(SZ),PBSPEED!(SZ),PBRADRNG!(SZ),PBENDUR(SZ),PBEFFECT(SZ),_
    PBACOST!(SZ),PBOCOST!(SZ),PBMCOST!(SZ),PBLIFE(SZ),PBHOME(SZ)

Print
Print "The current file of ship class names and related data will be "
Print "displayed on the following screens.  Press any key to continue"

While not INSTAT : WEND
      K$ = INKEY$
CLS

     OPEN "sclass" for input as #1
     y=1
     z=18

WHILE not EOF(1)     'print to screen one screen at a time

Print "List of All Ship Classes on File"
Print
 Print TAB(6) "Ship" TAB(17) "Max" TAB(23) "Radar" TAB(29) "Max"_
  TAB(36) "Effect-" TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life"
```

```
 Print TAB(6) "Class" TAB(17) "Speed" TAB(23) "Range" TAB(29) "Endur."_
  TAB(36) "iveness" TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"
 Print TAB(6) "Name" TAB(17) "(KTS)" TAB(23) "(NM)" TAB(29) "(hours)"_
  TAB(37) "(1-9)" TAB(45) "($)" TAB(51) "($/OH)" TAB(59) "($/OH)"_
   TAB(66) "(years)"
 Print

     FOR x = y to z

  INPUT #1, SCNAM$(x),PBSPEED!(x),PBRADRNG!(x),PBENDUR(x),PBEFFECT(x),_
    PBACOST!(x),PBOCOST!(x),PBMCOST!(x),PBLIFE(x)
  print x TAB(5) SCNAM$(x) TAB(17) PBSPEED!(x) TAB(22) PBRADRNG!(x)_
    TAB(29) PBENDUR(x) TAB(37) PBEFFECT(x) TAB(41) PBACOST!(x)_
    TAB(50) PBOCOST!(x) TAB(59) PBMCOST!(x) TAB(67) PBLIFE(x)

       If not EOF(1) then
        exit if
       else
        exit for
       End If

     NEXT x

     y = x
     z = x + 17

While not INSTAT : WEND
      K$ = INKEY$
     cls

LOOP
     CLOSE #1

For y = 1 to SZ     'specify number of coast guard patrol boats
 Print "Input line number, quantity, and home port number"
 Print "of each patrol boat class added "
 Input, L,Q,H
    If L <> 0 then
     exit if
    else
     exit for
    end if
     For x = 1 to Q
      Write #10, SCNAM$(L),PBSPEED!(L),PBRADRNG!(L),PBENDUR(L),_
      PBEFFECT(L),PBACOST!(L),PBOCOST!(L),PBMCOST!(L),PBLIFE(L),H
      Next x
Next y

Close #10

END SUB
```

```
SUB AOB

DIM ACNAM$(AZ),ASPEED!(AZ),ARADRNG!(AZ),AENDUR(AZ),_
    AACOST!(AZ),AOCOST!(AZ),AMCOST!(AZ),ALIFE(AZ),AHOME$(AZ)

Print "The current file of aircraft names and related data will be"
Print "displayed on the following screens.  Press any key to continue"

While not INSTAT : WEND
      K$ = INKEY$
CLS

     OPEN "ACLass" for input as #1
     y=1
     z=18

WHILE not EOF(1)     'print to screen one screen at a time

Print "List of All Aircraft Types on File"
Print
 Print TAB(6) "Aircraft" TAB(17) "Max" TAB(23) "Radar" TAB(29) "Max"_
  TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life"
 Print TAB(17) "Speed" TAB(23) "Range" TAB(29) "Endur."_
   TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"
 Print TAB(6) "Name" TAB(17) "(KTS)" TAB(23) "(NM)" TAB(29) "(hours)"_
   TAB(45) "($)" TAB(51) "($/OH)" TAB(59) "($/OH)"_
   TAB(66) "(years)"
 Print

     FOR x = y to z

       INPUT #1, ACNAM$(x),ASPEED!(x),ARADRNG!(x),AENDUR(x),_
       AACOST!(x),AOCOST!(x),AMCOST!(x),ALIFE(x)
       print x TAB(5) ACNAM$(x) TAB(17) ASPEED!(x) TAB(22) ARADRNG!(x)_
      TAB(29) AENDUR(x) TAB(41) AACOST!(x)_
       TAB(50) AOCOST!(x) TAB(59) AMCOST!(x) TAB(67) ALIFE(x)

       If not EOF(1) then
        exit if
       else
        exit for
       End If

     NEXT x

     y = x
     z = x + 17

While not INSTAT : WEND
      K$ = INKEY$

     cls
```

```
LOOP
     CLOSE #1

For y = 1 to AZ    'specify which aircraft are in maritime air patrol
 Print "Input the line number, quantity, and home base code"
 Print " of each aircraft added "
 Input, L,Q,H$
    If L <> 0 then
     exit if
    else
     exit for
    end if
      For x = 1 to Q
      Write #10, ACNAM$(L),ASPEED!(L),ARADRNG!(L),AENDUR(L),_
       AACOST!(L),AOCOST!(L),AMCOST!(L),ALIFE(L),H$
      Next x
Next y

Close #10

END SUB


SUB EOB

DIM RNAM$(RZ),RRADRNG!(RZ),_
    RACOST!(RZ),ROCOST!(RZ),RMCOST!(RZ),RLIFE(RZ),RHOME$(RZ)


Print "The current file of radar names and related data will be "
Print "displayed on the following screens.  Press any key to continue"

While not INSTAT : WEND
      K$ = INKEY$
CLS

     OPEN "RCLass" for input as #1
     y=1
     z=18

WHILE not EOF(1)      'print to screen one screen at a time

Print "List of All Radar Types on File"
Print
 Print TAB(6) "Radar" TAB(23) "Radar" _
  TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life"
 Print TAB(6) "" TAB(23) "Range" _
  TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"
 Print TAB(6) "Name" TAB(23) "(NM)" _
  TAB(45) "($)" TAB(51) "($/YR)" TAB(59) "($/YR)"_
   TAB(66) "(years)"
 Print
```

```
     FOR x = y to z

       INPUT #1, RNAM$(x),RRADRNG!(x),_
       RACOST!(x),ROCOST!(x),RMCOST!(x),RLIFE(x)
       print x TAB(5) RNAM$(x) TAB(22) RRADRNG!(x)_
      TAB(41) RACOST!(x)_
       TAB(50) ROCOST!(x) TAB(59) RMCOST!(x) TAB(67) RLIFE(x)

       If not EOF(1) then
        exit if
       else
        exit for
       End If

     NEXT x

     y = x
     z = x + 17

While not INSTAT : WEND
      K$ = INKEY$

     cls

LOOP
     CLOSE #1

For y = 1 to RZ  'specify which radars are in shore based radar force
 Print "Input the line number and home base code"
 Print "for each radar type added"
 Input, L,H$

    If L <> 0 then
     exit if
    else
     exit for
    end if
      Write #10, RNAM$(L),RRADRNG!(L),_
       RACOST!(L),ROCOST!(L),RMCOST!(L),RLIFE(L),H$
Next y

Close #10

END SUB


SUB DEPLOY

CLS

 Print
 Print "The percentage of ships on patrol can be varied."
 Print "Increasing the time on patrol increases the maintenance time"
 Print "proportionally.  Additionally, maintenance and operating"
```

```
 Print "costs rise proportionally."
 Print
 INPUT "Input percent of ships on patrol: ", SPATROL!
 INPUT "Input percent of aircraft on patrol: ", APATROL!

 Print
 Input "Input the ship patrol distance from shore (NM): "; DSPATROL!
 Input "Input the aircraft patrol distance from shore (NM): ";DAPATROL!

 Open DEPF$ for output as #1        'File this data for given country
  Write #1, SPATROL!,APATROL!,DSPATROL!,DAPATROL!
 Close #1

 CLS

END SUB


SUB THREAT


 Print
 Print
 Print "Do you want to:"
 Print "1. Modify Inbound Smuggling Threat File"
 Print "2. Modify Outbound Smuggling Threat File"
 Print "3. Modify Illegal Fishing Threat File"
 Print "4. Modify EEZ size"
 Print "5. Return to Main Menu"
 Print
 Print

444 INPUT "> ", THREATCASE

cls

 SELECT CASE THREATCASE

   CASE 1
     Print "This is the current Inbound Smugglers Threat File"
     Print
     OPEN INTHR$ for input as #10

 Print "Current Number of Smugglers" TAB(35) " Cost of Each "_
  TAB(55) " Max" TAB(65) "Alert"
 Print "        Each Day          " TAB(35) "to the Economy"_
  TAB(55) "Speed" TAB(65) "Dist."
 Print "Daylight          Night    " TAB(35) "  ($/boat)    "_
  TAB(55) "(KTS)" TAB(65) "(NM)"
 Print

 IF NOT EOF(10) THEN
  INPUT #10, IND!,INN!,INC!,INS!,INA!
  print IND! tab(20) INN! TAB(38) INC! TAB(56) INS! TAB(66) INA!
```

```
END IF
 CLOSE #10

Print
Print
Input "Do you want to change these values"; LOGI$
 If lcase$(left$(logi$,1)) = "y"  THEN
  OPEN INTHR$ for output as #10
  Input "Input values separated by commas: ",IND!,INN!,INC!,INS!,INA!
  Write #10, IND!,INN!,INC!,INS!,INA!
  Close #10
 END IF
cls


  CASE 2
    Print "This is the current Outbound Smugglers Threat File"
    Print
    OPEN OUTTHR$ for input as #10

Print "Current Number of Smugglers" TAB(35) " Cost of Each "_
 TAB(55) " Max" TAB(65) "Alert"
Print "        Each Day          " TAB(35) "to the Economy"_
 TAB(55) "Speed" TAB(65) "Dist."
Print "Daylight          Night   " TAB(35) "  ($/boat)    "_
 TAB(55) "(KTS)" TAB(65) "(NM)"
Print

IF NOT EOF(10) THEN
 INPUT #10, OUTD!,OUTN!,OUTC!,OUTS!,OUTA!
 print OUTD! tab(20) OUTN! TAB(38) OUTC! TAB(56) OUTS! TAB(66) OUTA!
END IF
 CLOSE #10
Print
Print
Input "Do you want to change these values"; LOGI$
 If lcase$(left$(logi$,1)) = "y"  THEN
  OPEN OUTTHR$ for output as #10
Input "Input values separated by commas: ",_
            OUTD!,OUTN!,OUTC!,OUTS!,OUTA!
  Write #10, OUTD!,OUTN!,OUTC!,OUTS!,OUTA!
  Close #10
 END IF
cls


  CASE 3
    Print "This is the current Illegal Fishing Threat File"
    Print
    OPEN FISHTHR$ for input as #10

Print "Current Number of Illegal  " TAB(28) "Average"_
 TAB(40) " Cost of Each " TAB(55) " Max" TAB(65) "Alert"
Print "    Fishermen Each Day     " TAB(28) "Distance"_
```

```
 TAB(40) "to the Economy" TAB(55) "Speed" TAB(65) "Dist."
 Print "Daylight          Night   " TAB(28) "Fm Shore"_
 TAB(40) "  ($/boat)    " TAB(55) "(KTS)" TAB(65) " (NM)"
 Print

 IF NOT EOF(10) THEN
  INPUT #10, FISHD!,FISHN!,FDST!,FISHC!,FISHS!,FISHA!
  print FISHD! tab(20) FISHN! TAB(30) FDST! TAB(42) FISHC!_
   TAB(56) FISHS! TAB (66) FISHA!
 END IF
  CLOSE #10
 Print
 Print
 Input "Do you want to change these values"; LOGI$
   If lcase$(left$(logi$,1)) = "y"  THEN
     OPEN FISHTHR$ for output as #10
Input "Input values separated by commas: ",FISHD!,FISHN!,FDST!,_
                                 FISHC!,FISHS!,FISHA!
Write #10, FISHD!,FISHN!,FDST!,FISHC!,FISHS!,FISHA!
     Close #10
   END IF
cls

   CASE 4
   Open EEZ$ for output as #1
   Print "What does the country claim for an exclusive economic zone?"
   Input "(the standard is 200 NM):  ", EEZ
   Write #1, EEZ
   Close #1

   CASE 5
     EXIT SELECT

   CASE ELSE
     PRINT "Enter a number 1 through 4"
     GOTO 444

 END SELECT

END SUB


SUB SCLASS

DIM SCNAM$(SZ),PBSPEED!(SZ),PBRADRNG!(SZ),PBENDUR(SZ),PBEFFECT(SZ),_
    PBACOST!(SZ),PBOCOST!(SZ),PBMCOST!(SZ),PBLIFE(SZ)


Print "The current file of ship class names and related data will be "
Print "displayed on the following screens.  Press any key to continue"

While not INSTAT : WEND
      K$ = INKEY$
CLS
```

```
     OPEN "sclass" for input as #1
     y=1
     z=18

WHILE not EOF(1)     'print to screen one screen at a time

Print "List of All Ship Classes on File"
Print
 Print TAB(6) "Ship" TAB(17) "Max" TAB(23) "Radar" TAB(29) "Max"_
  TAB(36) "Effect-" TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life"
 Print TAB(6) "Class" TAB(17) "Speed" TAB(23) "Range" TAB(29) "Endur."_
  TAB(36) "iveness" TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"
 Print TAB(6) "Name" TAB(17) "(KTS)" TAB(23) "(NM)" TAB(29) "(hours)"_
  TAB(37) "(1-9)" TAB(45) "($)" TAB(51) "($/OH)" TAB(59) "($/OH)"_
   TAB(66) "(years)"
 Print

     FOR x = y to z

       INPUT #1, SCNAM$(x),PBSPEED!(x),PBRADRNG!(x),PBENDUR(x),_
       PBEFFECT(x),PBACOST!(x),PBOCOST!(x),PBMCOST!(x),PBLIFE(x)
       print x TAB(5) SCNAM$(x) TAB(17) PBSPEED!(x)_
      TAB(22) PBRADRNG!(x) TAB(29) PBENDUR(x) TAB(37) PBEFFECT(x)_
       TAB(41) PBACOST!(x) TAB(50) PBOCOST!(x) TAB(59) PBMCOST!(x)_
        TAB(67) PBLIFE(x)

       If not EOF(1) then
        exit if
       else
        exit for
       End If

     NEXT x

     y = x
     z = x + 17

While not INSTAT : WEND
      K$ = INKEY$
     cls

LOOP
     CLOSE #1


      Print
      Print "                SHIP CLASS MENU"

     Print "1. Add a ship"
     Print "2. Delete a ship"
     Print "3. Return to Main Menu"
     Print
```

```
2100 INPUT "> ", SCLASSCASE

     SELECT CASE SCLASSCASE

     CASE 1   'add a ship

     OPEN "sclass" for append as #1

 Print TAB(6) "Ship" TAB(17) "Max" TAB(23) "Radar" TAB(29) "Max"_
  TAB(36) "Effect-" TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life"
 Print TAB(6) "Class" TAB(17) "Speed" TAB(23) "Range" TAB(29) "Endur."_
  TAB(36) "iveness" TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"
 Print TAB(6) "Name" TAB(17) "(KTS)" TAB(23) "(NM)" TAB(29) "(hours)"_
  TAB(37) "(1-9)" TAB(45) "($)" TAB(51) "($/OH)" TAB(59) "($/OH)"_
   TAB(66) "(years)"
 Print
 Print "Enter values separated by commas"

   x = x + 1
 INPUT SCNAM$(x),PBSPEED!(x),PBRADRNG!(x),PBENDUR(x),PBEFFECT(x),_
 PBACOST!(x),PBOCOST!(x),PBMCOST!(x),PBLIFE(x)

   CLOSE #1

   OPEN "SCLASS" for output as #1  'rebuild file SCLASS

    For i = 1 to x
     If SCNAM$(i) <> "" then
       Write #1, SCNAM$(i),PBSPEED!(i),PBRADRNG!(i),PBENDUR(i),_
       PBEFFECT(i),PBACOST!(i),PBOCOST!(i),PBMCOST!(i),PBLIFE(i)
     End If
    Next i

   CLOSE #1

     CASE 2      'delete a ship

     Input "Input the number of the ship class to be deleted ", DN
     SCNAM$(DN)= ""

   OPEN "SCLASS" for output as #1  'rebuild file SCLASS

    For i = 1 to x
     If SCNAM$(i) <> "" then
       Write #1, SCNAM$(i),PBSPEED!(i),PBRADRNG!(i),PBENDUR(i),_
       PBEFFECT(i),PBACOST!(i),PBOCOST!(i),PBMCOST!(i),PBLIFE(i)
     End If
    Next i

   CLOSE #1

     CASE 3
```

```
      EXIT SELECT
     CASE ELSE
      PRINT "Input a number 1-3"
      Goto 2100
     END SELECT

END SUB


SUB ATYPE

DIM ACNAM$(AZ),ASPEED!(AZ),ARADRNG!(AZ),AENDUR(AZ),_
    AACOST!(AZ),AOCOST!(AZ),AMCOST!(AZ),ALIFE(AZ)


Print "The current file of aircraft names and related data will be"
Print "displayed on the following screens.  Press any key to continue"

While not INSTAT : WEND
      K$ = INKEY$
CLS

     OPEN "AClass" for input as #1
     y=1
     z=18

WHILE not EOF(1)     'print to screen one screen at a time

Print "List of All Aircraft Types on File"
Print
 Print TAB(6) "Aircraft" TAB(17) "Max" TAB(23) "Radar" TAB(29) "Max"_
  TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life"
 Print TAB(17) "Speed" TAB(23) "Range" TAB(29) "Endur."_
   TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"
 Print TAB(6) "Name" TAB(17) "(KTS)" TAB(23) "(NM)" TAB(29) "(hours)"_
   TAB(45) "($)" TAB(51) "($/OH)" TAB(59) "($/OH)"_
   TAB(66) "(years)"
 Print

     FOR x = y to z

       INPUT #1, ACNAM$(x),ASPEED!(x),ARADRNG!(x),AENDUR(x),_
       AACOST!(x),AOCOST!(x),AMCOST!(x),ALIFE(x)
       print x TAB(5) ACNAM$(x) TAB(17) ASPEED!(x) TAB(22) ARADRNG!(x)_
      TAB(29) AENDUR(x) TAB(41) AACOST!(x)_
       TAB(50) AOCOST!(x) TAB(59) AMCOST!(x) TAB(67) ALIFE(x)

       If not EOF(1) then
        exit if
       else
        exit for
       End If
```

```
     NEXT x

     y = x
     z = x + 17

While not INSTAT : WEND
      K$ = INKEY$
     cls

LOOP
     CLOSE #1


      Print
      Print "                AIRCRAFT TYPE MENU"

     Print "1. Add an aircraft"
     Print "2. Delete an aircraft"
     Print "3. Return to Main Menu"
     Print

3100 INPUT "> ", ACLASSCASE

     SELECT CASE ACLASSCASE

     CASE 1   'add an aircraft

     OPEN "AClass" for append as #1

 Print TAB(6) "Aircraft" TAB(17) "Max" TAB(23) "Radar" TAB(29) "Max"_
  TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life"
 Print TAB(17) "Speed" TAB(23) "Range" TAB(29) "Endur."_
  TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"
 Print TAB(6) "Name" TAB(17) "(KTS)" TAB(23) "(NM)" TAB(29) "(hours)"_
  TAB(45) "($)" TAB(51) "($/OH)" TAB(59) "($/OH)"_
   TAB(66) "(years)"
 Print
 Print "Enter values separated by commas"

   x = x + 1
 INPUT ACNAM$(x),ASPEED!(x),ARADRNG!(x),AENDUR(x),AACOST!(x),_
       AOCOST!(x),AMCOST!(x),ALIFE(x)

   CLOSE #1

   OPEN "ACLASS" for output as #1  'rebuild file ACLASS

    For i = 1 to x
     If ACNAM$(i) <> "" then
       Write #1, ACNAM$(i),ASPEED!(i),ARADRNG!(i),AENDUR(i),_
       AACOST!(i),AOCOST!(i),AMCOST!(i),ALIFE(i)
     End If
    Next i
```

```
   CLOSE #1

     CASE 2      'delete an aircraft

     Input "Input the number of the aircraft to be deleted ", DN
     ACNAM$(DN)= ""

   OPEN "ACLASS" for output as #1  'rebuild file ACLASS

    For i = 1 to x
     If ACNAM$(i) <> "" then
       Write #1, ACNAM$(i),ASPEED!(i),ARADRNG!(i),AENDUR(i),_
       AACOST!(i),AOCOST!(i),AMCOST!(i),ALIFE(i)
     End If
    Next i

   CLOSE #1

     CASE 3
      EXIT SELECT
     CASE ELSE
      PRINT "Input a number 1-3"
      Goto 3100
     END SELECT

END SUB


SUB SRADAR

DIM RNAM$(RZ),RRADRNG!(RZ),_
    RACOST!(RZ),ROCOST!(RZ),RMCOST!(RZ),RLIFE(RZ)

Print "The current file of radar names and related data will be"
Print "displayed on the following screens.  Press any key to continue"

While not INSTAT : WEND
      K$ = INKEY$
CLS

     OPEN "RClass" for input as #1
     y=1
     z=18

WHILE not EOF(1)     'print to screen one screen at a time

Print "List of All Radar Types on File"
Print
 Print TAB(6) "Radar" TAB(23) "Radar" _
  TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life"
 Print TAB(6) "" TAB(23) "Range" _
  TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"
```

```
 Print TAB(6) "Name" TAB(23) "(NM)"_
  TAB(45) "($)" TAB(51) "($/YR)" TAB(59) "($/YR)"_
   TAB(66) "(years)"
 Print

     FOR x = y to z

       INPUT #1, RNAM$(x),RRADRNG!(x),_
       RACOST!(x),ROCOST!(x),RMCOST!(x),RLIFE(x)
       print x TAB(5) RNAM$(x) TAB(22) RRADRNG!(x)_
      TAB(41) RACOST!(x)_
       TAB(50) ROCOST!(x) TAB(59) RMCOST!(x) TAB(67) RLIFE(x)

       If not EOF(1) then
        exit if
       else
        exit for
       End If
     NEXT x

     y = x
     z = x + 17

While not INSTAT : WEND
      K$ = INKEY$
     cls

LOOP
     CLOSE #1

      Print
      Print "                RADAR TYPE MENU"

     Print "1. Add a radar"
     Print "2. Delete a radar"
     Print "3. Return to Main Menu"
     Print

4100 INPUT "> ", RCLASSCASE

     SELECT CASE RCLASSCASE

     CASE 1   'add a radar

     OPEN "RCLass" for append as #1

 Print TAB(6) "Radar"  TAB(23) "Radar"_
  TAB(45) "Acq." TAB(52) "Op" TAB(60) "Maint"_
   TAB(67) "Life"
 Print TAB(6) "" TAB(23) "Range"_
  TAB(45) "Cost" TAB(52) "Cost" TAB(60) "Cost"
 Print TAB(6) "Name" TAB(23) "(NM)"_
  TAB(45) "($)" TAB(51) "($/YR)" TAB(59) "($/YR)"_
   TAB(66) "(years)"
```

```
 Print
 Print "Enter values separated by commas"

   x = x + 1
 INPUT RNAM$(x),RRADRNG!(x),RACOST!(x),ROCOST!(x),RMCOST!(x),RLIFE(x)

   CLOSE #1

   OPEN "RCLASS" for output as #1  'rebuild file RCLASS

    For i = 1 to x
     If RNAM$(i) <> "" then
       Write #1, RNAM$(i),RRADRNG!(i),_
       RACOST!(i),ROCOST!(i),RMCOST!(i),RLIFE(i)
     End If
    Next i

   CLOSE #1

     CASE 2      'delete a radar

     Input "Input the number of the radar to be deleted ", DN
     RNAM$(DN)= ""

   OPEN "RCLASS" for output as #1  'rebuild file RCLASS

    For i = 1 to x
     If RNAM$(i) <> "" then
       Write #1, RNAM$(i),RRADRNG!(i),_
       RACOST!(i),ROCOST!(i),RMCOST!(i),RLIFE(i)
     End If
    Next i

   CLOSE #1

     CASE 3
      EXIT SELECT
     CASE ELSE
      PRINT "Input a number 1-3"
      Goto 4100
     END SELECT

END SUB

$SEGMENT               'ALLOCATES 64k MORE MEMORY

SUB SIMULATE

DIM PLON(PZ),PLAT(PZ),ALON(AFZ),ALAT(AFZ),RLAT(RZ),RLON(RZ),SPLON(30),_
    APLON(30),CLON(30),SDEPAGE(SZ),ADEPAGE(SZ)
DIM COAST$(3),TBLOCKS!(SZ),TBLOCKA!(AZ),INLAT(50),INLON(50),_
    BACKS(SZ),BACKA(AZ),TBLOCKIN!(50),TBLOCKOUT!(50),TBLOCKFISH!(50)
DIM OUTLAT(50),OUTLON(50),FISHLAT(50),FISHLON(50),EEZEDGE(25)
DIM PBSPEED!(SZ),PBRADRNG!(SZ),PBENDUR(SZ),PBEFFECT(SZ),PBHOME(SZ)
```

```
DIM ASPEED!(AZ),ARADRNG!(AZ),AENDUR(AZ),AHOME$(AZ)
DIM RRADRNG!(RZ),RHOME$(RZ)

 BLOCKSIZE! = (2.8E-6)*SCALE!   'size of one screen block in NM

  'Retrieve deployment data

 Open DEPF$ for input as #1
  Input #1, SPATROL!,APATROL!,DSPATROL!,DAPATROL!
 Close #1

   'Retrieve threat data

 OPEN INTHR$ for input as #1
  Input #1, IND!,INN!,INC!,INS!,INA!
 Close #1

  OPEN OUTTHR$ for input as #1
  Input #1, OUTD!,OUTN!,OUTC!,OUTS!,OUTA!
 Close #1

 OPEN FISHTHR$ for input as #1
  Input #1, FISHD!,FISHN!,FDST!,FISHC!,FISHS!,FISHA!
 Close #1

 Open EEZ$ for input as #1
   Input #1, EEZ
 Close #1

For x = 1 to 24       'Find locations of ports, airfields, and radars
   For y = 1 to 79
   s$=lcase$(layout$(x,y))

   select case s$

   case "l"
    exit select

   case "w"
    exit select

   case "1" TO "9"             'Ports
    PLAT(val(s$)) = x       'equates the x grid to LATitude
    PLON(val(s$)) = y       'equates the y grid to LONgitude

   case "a" TO "i"             'Airfields
    ALAT(ascii(s$)-96) = x
    ALON(ascii(s$)-96) = y

   case  "m" TO "u"             'RADAR stations
    RLAT(ASCII(S$)-108) = x
    RLON(ASCII(S$)-108) = y
    exit select
```

```
   case else
    Print  "Bad character in coastal area layout"

   END SELECT
  Next y
 Next x

   'Find shoreline and assign Patrol areas

For x = 1 to 24
  For y = 1 to 79

   j =(y) mod 2       'Toggles 1 and 0 to compare previous values
   m =(y+1) mod 2       ' Previous subscript


    Coast$(j)=lcase$(layout$(x,y))

      BEACHCASE$= Coast$(m) + Coast$(j)

       SELECT CASE BEACHCASE$

      CASE "l1" to "lu"            'other interfaces (not beach)
        Exit Select

      CASE "l"                     'END OF LINE
        EXIT SELECT

      CASE "w1" to "wu"      'water on the left
       EDGE = 1                  'DEFINES WATER ON LEFT

       SPLON(x) = y - DSPATROL!/BLOCKSIZE!       'ship patrol boxes

       APLON(x) = y - DAPATROL!/BLOCKSIZE!       'air patrol boxes

       CLON(x) = y                             'Edge of coast
       EEZEDGE(x) = CLON(x) - INT(EEZ/BLOCKSIZE!)
        IF EEZEDGE(x) > EDGE THEN EEZEDGE(x) = EDGE

       Exit Select

      CASE "lw"                        'water on the right

       EDGE = 47                       'DEFINES WATER ON RIGHT

       SPLON(x) = y + DSPATROL!/BLOCKSIZE!

       APLON(x) = y + DAPATROL!/BLOCKSIZE!

       CLON(x) = y                             'Edge of coast
       EEZEDGE(x) = CLON(x) + INT(EEZ/BLOCKSIZE!)
        IF EEZEDGE(x) > EDGE THEN EEZEDGE(x) = EDGE

       EXIT SELECT
```

```
      CASE "lw" to "uw"                  'water on the right
         CHECK1$= left$(lcase$(BEACHCASE$),1)
      If CHECK1$ = "l" then    'eliminates other interfaces (eg. 12
           exit select
      Else
         CHECK2$= mid$(lcase$(BEACHCASE$),2,1)
       If CHECK2$ <> "l" then  'eliminates other interfaces (eg. al
         EDGE = 47                        'DEFINES WATER ON RIGHT

       SPLON(x) = y + DSPATROL!/BLOCKSIZE!

       APLON(x) = y + DAPATROL!/BLOCKSIZE!

       CLON(x) = y                               'Edge of coast
       EEZEDGE(x) = CLON(x) + INT(EEZ/BLOCKSIZE!)
        IF EEZEDGE(x) > EDGE THEN EEZEDGE(x) = EDGE

      End If
       End If
      CASE else
       EXIT SELECT

       END SELECT
   Next y
   Coast$(0)= ""
   Coast$(1)= ""           'reset at end of line
 Next x


    'Order of Battle input

 OPEN NOBF$ for input as #10

p = 0

WHILE not EOF(10)
p = p + 1
  INPUT #10, SCNAM$,PBSPEED!(p),PBRADRNG!(p),PBENDUR(p),PBEFFECT(p),_
    PBACOST!,PBOCOST!,PBMCOST!,PBLIFE,PBHOME(p)

LOOP
 CLOSE #10

 OPEN AOBF$ for input as #10
a = 0

WHILE not EOF(10)
a = a + 1

    INPUT #10, ACNAM$,ASPEED!(a),ARADRNG!(a),AENDUR(a),_
     AACOST!,AOCOST!,AMCOST!,ALIFE,AHOME$(a)

LOOP
```

```
 CLOSE #10

 MTIMER

 OPEN EOBF$ for input as #10
e = 0

WHILE not EOF(10)
e = e + 1

   INPUT #10, RNAM$,RRADRNG!(e),_
    RACOST!,ROCOST!,RMCOST!,RLIFE,RHOME$(e)

LOOP
 CLOSE #10

   'Choose the initial ships and aircraft on patrol and standby

   Randomize MTIMER

     NUMSDEP = INT((SPATROL!/100) * p) + 1 'number of ships deployed
     NUMADEP = INT((APATROL!/100) * a) + 1 'number of aircraft deployed

     NUMSSTBY = p - (2 * NUMSDEP)  'number of ships on stby

       x = 0
     DO UNTIL x >= NUMSDEP  'Randomly picks deployed ships w/o duplic.
       x = x + 1
      SDEP(x) = INT(RND * p) + 1
          For y = 1 to (x-1)
           If SDEP(y) = SDEP(x) then
            x = x - 1
           End If
          Next y
     Loop

       x = 0
      For i = 1 to p
       x = x + 1
       If x > NUMSSTBY then Exit For
      SSTBY(x) = i
          For z = 1 to NUMSDEP
            If SDEP(z) = SSTBY(x) then   'check if deployed
              x = x - 1
             Exit for
            End If
          Next z
       Next i

     NUMSMAINT = p - NUMSDEP - NUMSSTBY

       x = 0
    For s = 1 to p
       x = x + 1
```

```
    If x > NUMSMAINT THEN EXIT FOR
  SMAINT(x) = s
   FOR n = 1 TO (NUMSDEP + NUMSSTBY)
    IF (SMAINT(x) = SSTBY(n)) OR (SMAINT(x) = SDEP(n)) THEN
   x = x - 1
   EXIT FOR
    END IF
   NEXT n
 NEXT s


    x = 0
  DO UNTIL x >= NUMADEP  'Randomly picks deployed aircraft
    x = x + 1
   ADEP(x) = INT(RND * a) + 1
       For y = 1 to (x-1)
        If ADEP(y) = ADEP(x) then
         x = x - 1
        End If
       Next y
  Loop

  NUMAMAINT = a - NUMADEP

    x = 0
 For s = 1 to a
    x = x + 1
    If x > NUMAMAINT THEN EXIT FOR
  AMAINT(x) = s
   FOR n = 1 TO NUMADEP
    IF AMAINT(x) = ADEP(n) THEN
   x = x - 1
   EXIT FOR
    END IF
   NEXT n
 NEXT s


'Allocate Patrol Boxes

    'Ships

    BOXPERSHIP = 24\NUMSDEP
    x = BOXPERSHIP\2                'spaces ships evenly
   For s = 1 to NUMSDEP
       SHIPLON(SDEP(s)) = SPLON(x)         'LON of deployed ship
       SHIPLAT(SDEP(s)) = x                'LAT of deployed ship
       x = x + BOXPERSHIP
   Next s

   For x = 1 to NUMSSTBY

       SHIPLON(SSTBY(x)) = PLON(PBHOME(SSTBY(x)))  'LON of stby
       SHIPLAT(SSTBY(x)) = PLAT(PBHOME(SSTBY(x)))  'LAT of stby
```

```
    Next x

'Aircraft

     BOXPERACFT = 24\NUMADEP

     x = BOXPERACFT\2                    'SPACES AIRCRAFT EVENLY
    For y = 1 to NUMADEP
        ACFTLON(ADEP(y)) = APLON(x)          'LON of deployed acft
        ACFTLAT(ADEP(y)) = x                'LAT of deployed acft
        x = x + BOXPERACFT
    Next y


    For s = 1 to p    'Time for ship to move one block

     TBLOCKSO!(s) = 10 * BLOCKSIZE!/(PBSPEED!(s)/2)
     TBLOCKS!(s)  = TBLOCKSO!(s)
    Next s

    For s = 1 to a    'Time for aircraft to move one block

     TBLOCKAO!(s) = 10 * BLOCKSIZE!/(ASPEED!(s))
     TBLOCKA!(s) = TBLOCKAO!(s)

    Next s

     'Time for threat contacts to move one block

     TBLOCKINO! = 10 * BLOCKSIZE!/INS!
     TBLOCKOUTO! = 10 * BLOCKSIZE!/OUTS!
     TBLOCKFISHO! = 10 * BLOCKSIZE!/FISHS!
     'Initialize timer counters

For s = 1 to 50
     TBLOCKIN!(s) = TBLOCKINO!
     TBLOCKOUT!(s) = TBLOCKOUTO!
     TBLOCKFISH!(s) = TBLOCKFISHO!
Next s

 'INITIAL RADAR MATRIX

 ID$ = "s"
CALL RADARMATRIX(RLAT(),RLON(),PBRADRNG!(),ARADRNG!(),RRADRNG!(),e,ID$)
 'RADAR coverage matrix

 ID$ = "a"
CALL RADARMATRIX(RLAT(),RLON(),PBRADRNG!(),ARADRNG!(),RRADRNG!(),e,ID$)
 'RADAR coverage matrix

 ID$ = "r"
CALL RADARMATRIX(RLAT(),RLON(),PBRADRNG!(),ARADRNG!(),RRADRNG!(),e,ID$)
 'RADAR coverage matrix
```

```
Input "Do you want to see the radar coverage"; logi$
     lo$= lcase$(left$(logi$,1))
   If lo$ = "y" then
cls
 for x=1 to 24
   PRINT
  for y = 1 to 79
       R$ = LEFT$(RADAR$(x,y),1)
     Select Case R$
      CASE "R"
      Color 7,6,0
      Print " ";
      CASE Else
      Color 7,1,0
      Print " ";
    END SELECT
   next y
 next x

 while not instat : wend
      K$ = INKEY$

 color 7,1,8
 cls
  END IF


   'BASIC TIMING SIMULATION
INPUT "Input number of days to let simulation stabilize: ",STABILIZE
INPUT "Input number of days to run simulation after it is stabile: ",_
       DAYS

       r! = RND(1)
       COMPLI! = (LOG(1-r!) + LOG(r!))/2   'COMPLEMENTARY RANDOM PAIR
       COMPLO! = COMPLI!
       COMPLF! = COMPLI!

   OPEN "TSTTST" FOR OUTPUT AS #9

FOR d = 1 to ((DAYS+STABILIZE)*2)   'NUMBER OF half days
  For t = 1 to 120   '6 minute intervals for one HALF day

   ELT = ((d-1)*120)+t

  'SHIP MOVEMENT SEQUENCE

      For s = 1 to NUMSDEP

If SREST(SDEP(s)) = 0 THEN 'freezes overworked ships

       SDEPAGE(s) = SDEPAGE(s) + 1
       SOPHOURS!(SDEP(s)) = SOPHOURS!(SDEP(s)) + 0.1

      If TBLOCKS!(SDEP(s)) < ELT then
```

```
     TBLOCKS!(SDEP(s))= TBLOCKS!(SDEP(s)) + TBLOCKSO!(SDEP(s))

      x = SHIPLAT(SDEP(s))
      y = SPLON(x)
'UPDATE RADAR BOX (REMOVES OLD COVERAGE)
    RNG! = PBRADRNG!(SDEP(s))/BLOCKSIZE!
    XMIN = x-RNG!
    XMAX = x+RNG!
    YMIN = y-RNG!
    YMAX = y+RNG!
   For j = XMIN to XMAX
    For k = YMIN to YMAX

    DISTANCE! =  (SQR(((j-x)^2) + ((k-y)^2)))

    IF DISTANCE! < RNG! THEN
      RADAR$(j,k) = MID$(RADAR$(j,k),2)

    End If

    Next k
  Next j

                    'REPLACES SHIPS
       IF SDEPAGE(s) > 10 * PBENDUR(SDEP(s)) THEN
         CALL REPLACESHIP(s,PLAT(),PLON(),PBHOME(),PBSPEED!())
         SDEPAGE(s) = 0
         TBLOCKS!(SDEP(s)) = ELT

       END IF

    If SHIPLAT(SDEP(s)) => 24 then  BACKS(s) = 1
    If BACKS(s) = 0 THEN         'move to next box
     SHIPLAT(SDEP(s)) = SHIPLAT(SDEP(s)) + 1
    Else                       'COUNT BACKWARDS
     SHIPLAT(SDEP(s)) = SHIPLAT(SDEP(s)) - 1
     IF SHIPLAT(SDEP(s)) <= 1 THEN BACKS(s) = 0   'count up again
    End If
      x = SHIPLAT(SDEP(s))
      y = SPLON(x)
       SHIPLON(SDEP(s)) = y        'LON of deployed ship
    'UPDATE RADAR BOX
    XMIN = x-RNG!
    XMAX = x+RNG!
    YMIN = y-RNG!
    YMAX = y+RNG!
   For j = XMIN to XMAX
    For k = YMIN to YMAX

    DISTANCE! =  (SQR(((j-x)^2) + ((k-y)^2)))

    IF DISTANCE! < RNG! THEN
      RADAR$(j,k) =  RADAR$(j,k) + "R"
```

```
      End If

      Next k
    Next j

     END IF
END IF
     Next s

 'AIRCRAFT MOVEMENT SEQUENCE

     For s = 1 to NUMADEP

IF AREST(ADEP(s)) = 0 THEN 'freezes overworked acft

      ADEPAGE(s) = ADEPAGE(s) + 1
      AOPHOURS!(ADEP(s)) = AOPHOURS!(ADEP(s)) + 0.1

     If TBLOCKA!(ADEP(s)) < ELT then
       TBLOCKA!(ADEP(s))= TBLOCKA!(ADEP(s)) + TBLOCKAO!(ADEP(s))
        x = ACFTLAT(ADEP(s))
        y = APLON(x)         'LON of deployed AIRCRAFT

  'UPDATE RADAR BOX (REMOVES OLD COVERAGE)

     RNG! = ARADRNG!(ADEP(s))/BLOCKSIZE!
      XMIN = x-RNG!
      XMAX = x+RNG!
      YMIN = y-RNG!
      YMAX = y+RNG!
     For j = XMIN to XMAX
      For k = YMIN to YMAX

      DISTANCE! =  (SQR(((j-x)^2) + ((k-y)^2)))

      IF DISTANCE! < RNG! THEN
        RADAR$(j,k) = MID$(RADAR$(j,k),2)

      End If

      Next k
    Next j

                     'REPLACES ACFT
         IF ADEPAGE(s) > 10 * AENDUR(ADEP(s)) THEN
           CALL REPLACEACFT(s,ALAT(),ALON(),AHOME$())
           ADEPAGE(s) = 0
               TBLOCKA!(ADEP(s)) = ELT
           END IF

      If ACFTLAT(ADEP(s)) > 20 then  BACKA(s) = 1
      If BACKA(s) = 0 THEN        'move to next box
   ACFTLAT(ADEP(s)) = ACFTLAT(ADEP(s)) + (1/TBLOCKAO!(ADEP(s)))
      Else
```

```
   ACFTLAT(ADEP(s)) = ACFTLAT(ADEP(s)) - (1/TBLOCKAO!(ADEP(s)))
       IF ACFTLAT(ADEP(s)) < 4 THEN BACKA(s) = 0   'count up again
     End If
       x = ACFTLAT(ADEP(s))
       y = APLON(x)          'LON of deployed AIRCRAFT
       ACFTLON(ADEP(s)) = y
  'UPDATE RADAR BOX

     XMIN = x-RNG!
     XMAX = x+RNG!
     YMIN = y-RNG!
     YMAX = y+RNG!
    For j = XMIN to XMAX
     For k = YMIN to YMAX

     DISTANCE! =  (SQR(((j-x)^2) + ((k-y)^2)))

     IF DISTANCE! < RNG! THEN
       RADAR$(j,k) =  RADAR$(j,k) + "R"

     End If

     Next k
   Next j


    END IF
END IF
    Next s

'GENERATE RANDOM THREATS (exponential interarrival times)

  RANDOMIZE ELT

  If LIGHT$ = "D"  THEN                ' TELLS DAY FROM NIGHT
    INARR! = IND!/120
  ELSE
    INARR! = INN!/120
  END IF

  TIN! = TINCHECK! - ((1/INARR!)*COMPLI!)    'exponential distribution
    IF TIN! < ELT THEN
     TINCHECK! = ELT
      r! = RND(1)
      COMPLI! = (LOG(1-r!) + LOG(r!))/2   'COMPLEMENTARY RANDOM PAIR
     INTHREAT = INTHREAT + 1                  'CURRENT THREATS
     INTOT = INTOT + 1                        'TOTAL INBOUND THREATS
     INLAT(INTHREAT) = INT(RND(24)) + 1
     x = INLAT(INTHREAT)
     INLON(INTHREAT) = EEZEDGE(x)
     y = INLON(INTHREAT)
     TBLOCKIN!(INTHREAT) = ELT
    End if
```

```
If LIGHT$ = "D"  THEN
  OUTARR! = OUTD!/120
ELSE
  OUTARR! = OUTN!/120
END IF


TOUT! = TOUTCHECK!-((1/OUTARR!)*COMPLO!)
  IF TOUT! < ELT THEN
   TOUTCHECK! = ELT
     r! = RND(1)
     COMPLO! = (LOG(1-r!) + LOG(r!))/2
    OUTTHREAT = OUTTHREAT + 1
    OUTTOT = OUTTOT + 1
    OUTLAT(OUTTHREAT) = INT(RND(24)) + 1
    x = OUTLAT(OUTTHREAT)
    OUTLON(OUTTHREAT) = CLON(x)          'LON OF COAST
    y = OUTLON(OUTTHREAT)
    TBLOCKOUT!(OUTTHREAT) = ELT
  End if


If LIGHT$ = "D"  THEN
  FISHARR! = FISHD!/120
ELSE
  FISHARR! = FISHN!/120
END IF

TFISH! = TFISHCHECK! - ((1/FISHARR!)*COMPLF!)
  IF TFISH! < ELT THEN
   TFISHCHECK! = ELT
     r! = RND(1)
     COMPLF! = (LOG(1-r!) + LOG(r!))/2
    FISHTHREAT = FISHTHREAT + 1
    FISHTOT = FISHTOT + 1
    FISHLAT(FISHTHREAT) = INT(RND(24)) + 1
    x = FISHLAT(FISHTHREAT)
     IF EDGE > 2 THEN
     FISHLON(FISHTHREAT) = CLON(x) + FDST!/BLOCKSIZE!
     ELSE
     FISHLON(FISHTHREAT) = CLON(x) - FDST!/BLOCKSIZE!
     END IF
    y = FISHLON(FISHTHREAT)
    TBLOCKFISH!(FISHTHREAT) = ELT
  End if


  ' MOVE THREAT TARGETS

   'MOVE INBOUND SMUGGLERS
     LT$ = ""
For s = 1 to INTHREAT
    SAFE = 0
```

```
    If TBLOCKIN!(s) < ELT then
      TBLOCKIN!(s) = TBLOCKIN!(s) + TBLOCKINO! 'time to move
     x = INLAT(s)
     y = INLON(s)
      If EDGE > 2 THEN
        INLON(s) = INLON(s) - 1    'LON of smuggler from left
        IF INLON(s) <= CLON(x) THEN SAFE = 1  'reached safety
      else
        INLON(s) = INLON(s) + 1    'LON of smuggler from right
        IF INLON(s) >= CLON(x) THEN SAFE = 1 'reached safety
      END IF
    END IF
      IF SAFE > 0 THEN  'reached safety
        LT$ = "i"
        INLON(s) = 0
        INLAT(s) = 0
      END IF
     x = INLAT(s)
     y = INLON(s)
  If left$(RADAR$(x,y),1) = "R" THEN
      CALL INTERCEPT(x,y,INS!,INA!,_
 PLAT(),PLON(),CLON(),EEZEDGE(),PBEFFECT(),PBSPEED!(),PBHOME())
      SELECT CASE TPT

      CASE -1    'intercept not possible
        LT$ = "i"
        INLAT(s) = 0
        INLON(s) = 0

      CASE 0    'intercept certain
        LT$ = "i"
        INLAT(s) = 0
        INLON(s) = 0
        ININCPTTOT = ININCPTTOT + 1
        TBLOCKS!(SDEP(TPT1)) = ELT
      CASE ELSE    'intercept possible
        IF INLAT(s) < SHIPLAT(SDEP(TPT)) THEN
         BACKS(TPT) = 1                    'MOVE TOWARD THREAT
        ELSE
         BACKS(TPT) = 0
        END IF
                                   'max speed
   TBLOCKSO!(SDEP(TPT)) = 10*BLOCKSIZE!/PBSPEED!(SDEP(TPT))
      END SELECT
  END IF
Next s
  If LT$ <> "" then  GOSUB LESSTHREAT

 'Outbound smugglers
     LT$ = ""
For s = 1 to OUTTHREAT
    SAFE = 0
    If TBLOCKOUT!(s) < ELT then
 TBLOCKOUT!(s) = TBLOCKOUT!(s) + TBLOCKOUTO!
```

```
      x = OUTLAT(s)
      y = OUTLON(s)
       If EDGE > 2 THEN
   OUTLON(s) = OUTLON(s) + 1   'LON of smuggler with water on left
       IF OUTLON(s) > EEZEDGE(x) THEN SAFE = 1
       else
   OUTLON(s) = OUTLON(s) - 1   'LON of smuggler with water on right
       IF OUTLON(s) < EEZEDGE(x) THEN SAFE = 1
       END IF
     END IF
      IF SAFE > 0 THEN         'reached safety
         LT$ = "o"
         OUTLON(s) = 0
         OUTLAT(s) = 0
       END IF
      x = OUTLAT(s)
      y = OUTLON(s)
    If left$(RADAR$(x,y),1) = "R" THEN
      CALL INTERCEPT(x,y,OUTS!,OUTA!,_
PLAT(),PLON(),CLON(),EEZEDGE(),PBEFFECT(),PBSPEED!(),PBHOME())
       SELECT CASE TPT

       CASE -1     'intercept not possible
         LT$ = "o"
         OUTLAT(s) = 0
         OUTLON(s) = 0

       CASE 0     'intercept certain
         LT$ = "o"
         OUTLAT(s) = 0
         OUTLON(s) = 0
         OUTINCPTTOT = OUTINCPTTOT + 1
         TBLOCKS!(SDEP(TPT1)) = ELT
       CASE ELSE    'intercept possible
         IF OUTLAT(s) < SHIPLAT(SDEP(TPT)) THEN
          BACKS(TPT) = 1                     'MOVE TOWARD THREAT
         ELSE
          BACKS(TPT) = 0
         END IF
                                          'max speed
     TBLOCKSO!(SDEP(TPT)) = 10*BLOCKSIZE!/PBSPEED!(SDEP(TPT))
       END SELECT
    END IF
  Next s
    If LT$ <> "" then  GOSUB LESSTHREAT

  'Move illegal fishermen
      LT$ = ""
  For s = 1 to FISHTHREAT
     SAFE = 0
     If TBLOCKFISH!(s) < ELT then
    TBLOCKFISH!(s) = TBLOCKFISH!(s) + TBLOCKFISHO!
      x = FISHLAT(s)
      y = FISHLON(s)
```

```
          RANDOMMOVE = INT(RND(4)) + 1
       SELECT CASE RANDOMMOVE
          CASE 1
         FISHLON(s) = FISHLON(s) + 1
          CASE 2
         FISHLON(s) = FISHLON(s) - 1
          CASE 3
         FISHLAT(s) = FISHLAT(s) + 1
          CASE 4
         FISHLAT(s) = FISHLAT(s) - 1
          CASE ELSE
           EXIT SELECT
         END SELECT

     END IF
       IF (FISHLON(s) > EEZEDGE(x)) AND (EDGE > 2) THEN SAFE = 1
       IF (FISHLON(s) < EEZEDGE(x)) AND (EDGE < 2) THEN SAFE = 1
       IF SAFE > 0 THEN
           'reached safety
         LT$ = "f"
         FISHLON(s) = 0
         FISHLAT(s) = 0
       END IF
      x = FISHLAT(s)
      y = FISHLON(s)
   If left$(RADAR$(x,y),1) = "R" THEN
     CALL INTERCEPT(x,y,FISHS!,FISHA!,_
PLAT(),PLON(),CLON(),EEZEDGE(),PBEFFECT(),PBSPEED!(),PBHOME())
       SELECT CASE TPT

       CASE -1     'intercept not possible
         LT$ = "f"
         FISHLAT(s) = 0
         FISHLON(s) = 0

       CASE 0     'intercept certain
         LT$ = "f"
         FISHLAT(s) = 0
         FISHLON(s) = 0
         FISHINCPTTOT = FISHINCPTTOT + 1
         TBLOCKS!(SDEP(TPT1)) = ELT

       CASE ELSE    'intercept possible
         IF FISHLAT(s) < SHIPLAT(SDEP(TPT)) THEN
          BACKS(TPT) = 1                     'MOVE TOWARD THREAT
         ELSE
          BACKS(TPT) = 0
         END IF
                                         'max speed
    TBLOCKSO!(SDEP(TPT)) = 10*BLOCKSIZE!/PBSPEED!(SDEP(TPT))
       END SELECT
   END IF
  Next s
   If LT$ <> "" then  GOSUB LESSTHREAT
```

```
 Next t

   IF LIGHT$ = "D" THEN         'TURN DAY INTO NIGHT
    LIGHT$ = "N"
   ELSE
    LIGHT$ = "D"
   END IF

     DAYC! = d/2
   IF DAYC! > STABILIZE THEN
     IF (INTOT - INTHREAT) > 0 THEN
      MIEISMUG! = ININCPTTOT/(INTOT - INTHREAT)
     ELSE
      MIEISMUG! = 0
     END IF

     IF (OUTTOT - OUTTHREAT) > 0 THEN
      MIEOSMUG! = OUTINCPTTOT/(OUTTOT - OUTTHREAT)
     ELSE
      MIEOSMUG! = 0
     END IF

     IF (FISHTOT - FISHTHREAT) > 0 THEN
      MIEFISH! = FISHINCPTTOT/(FISHTOT - FISHTHREAT)
     ELSE
      MIEFISH! = 0
     END IF

     WRITE #9, DAYC! - STABILIZE,MIEISMUG!,MIEOSMUG!,MIEFISH!
   ELSE
     INTOT = INTHREAT
     OUTTOT = OUTTHREAT
     FISHTOT = FISHTHREAT
     ININCPTTOT = 0
     OUTINCPTTOT = 0
     FISHINCPTTOT = 0
     PATINCPTTOT = 0
     STBYINCPTTOT = 0
   END IF

   REDIM AREST(AZ),SREST(SZ)     'restarts resting ships and aircraft
Next d

CLOSE #9

EXIT SUB


LESSTHREAT:

SELECT CASE LT$

  CASE "i"
```

```
    z = INTHREAT
 For i = 1 to z
     IF INTHREAT <= 0 then exit for
     IF z > INTHREAT then exit for
  IF INLAT(i) = 0 THEN                  'safe or intercepted ship
   INTHREAT = INTHREAT - 1
    For j = i to INTHREAT
    INLAT(j) = INLAT(j+1)              'renumbers remaining threats
    INLON(j) = INLON(j+1)
    TBLOCKIN!(j) = TBLOCKIN!(j+1)
    Next j
   i = i - 1
  END IF
Next i


CASE "o"
     z = OUTTHREAT
 For i = 1 to z
     IF OUTTHREAT <= 0 then exit for
     IF z > OUTTHREAT then exit for
  IF OUTLAT(i) = 0 THEN
   OUTTHREAT = OUTTHREAT - 1
    For j = i to OUTTHREAT
    OUTLAT(j) = OUTLAT(j+1)
    OUTLON(j) = OUTLON(j+1)
    TBLOCKOUT!(j) = TBLOCKOUT!(j+1)
    Next j
   i = i - 1
  END IF
Next i


CASE "f"
     z = FISHTHREAT
 For i = 1 to z
     IF FISHTHREAT <= 0 then exit for
     IF z > FISHTHREAT then exit for
  IF FISHLAT(i) = 0 THEN
   FISHTHREAT = FISHTHREAT - 1
    For j = i to FISHTHREAT
    FISHLAT(j) = FISHLAT(j+1)
    FISHLON(j) = FISHLON(j+1)
    TBLOCKFISH!(j) = TBLOCKFISH!(j+1)
    Next j
   i = i - 1
  END IF
Next i

CASE ELSE
  EXIT SELECT

END SELECT
```

```
 RETURN

END SUB


SUB INTERCEPT(x,y,TSPEED!,ALERTD!,_
PLAT(),PLON(),CLON(),EEZEDGE(),PBEFFECT(),PBSPEED!(),PBHOME())

DIM TPOSS!(SZ)

 CMAX = 0
 TPT = 1

 'MEASURE DISTANCE AND TIME OF VIOLATOR FROM SAFETY

       COASTD = ABS(y - CLON(x))
       EEZD = ABS(y - EEZEDGE(x))

       IF EEZD < COASTD THEN
      SAFED = EEZD
       ELSE
      SAFED = COASTD
       END IF

       TSAFE! = BLOCKSIZE!*SAFED/TSPEED!

 'TRY TO INTERCEPT VIOLATOR WITH PATROL BOAT DEPLOYED

 FOR s = 1 to NUMSDEP

If SREST(SDEP(s)) = 0 THEN 'freezes overworked ships

  j = SHIPLAT(SDEP(s))
  k = SHIPLON(SDEP(s))
                'INITIAL SEPARATION DISTANCE
  ISEPD! =  (BLOCKSIZE! * SQR(((j-x)^2) + ((k-y)^2)))

             'TIME TO REACH THE ALERT RADIUS

  TALERT! =  (ISEPD! - ALERTD!)/PBSPEED!(SDEP(s))

             'TIME TO INTERCEPT AFTER ALERTMENT

TINTERCEPT! =(ALERTD!-PBEFFECT(SDEP(s)))/(PBSPEED!(SDEP(s))-TSPEED!)

 IF (TSPEED!>PBSPEED!(SDEP(s))) AND (ALERTD!>PBEFFECT(SDEP(s))) THEN
     IF TPOSS!(CMAX) = 0 THEN TPT = -1    'no intercept possible

   ELSE                                       'possible intercept
       TPOSS!(s) = TALERT! + TINTERCEPT!
       IF CMAX = 0 THEN CMAX = s
       IF TPOSS!(CMAX) >= TPOSS!(s) THEN
      TPT = s
      CMAX = s
```

```
      END IF
 END IF
END IF
 NEXT s

IF (TSAFE!>TPOSS!(CMAX)) AND (TPT > 0) AND (TPOSS!(CMAX) > 0) THEN
            'certain intercept
     s = TPT
          CALL REPLACESHIP(s,PLAT(),PLON(),PBHOME(),PBSPEED!())
      TPT1 = TPT
      TPT = 0
      PATINCPTTOT = PATINCPTTOT + 1
      EXIT SUB
 ELSE


 'TRY TO INTERCEPT VIOLATOR WITH PATROL BOAT ON STANDBY

 CMAX = 0
 STBYINCPT = -1

 FOR s = 1 to NUMSSTBY
  j = SHIPLAT(SSTBY(s))
  k = SHIPLON(SSTBY(s))
               'INITIAL SEPARATION DISTANCE
 ISEPD! =  (BLOCKSIZE! * SQR(((j-x)^2) + ((k-y)^2)))

       'TIME TO REACH THE ALERT RADIUS (add .5 hours since stby)
 TALERT! =  0.5 + (ISEPD! - ALERTD!)/PBSPEED!(SSTBY(s))

             'TIME TO INTERCEPT AFTER ALERTMENT
TINTERCEPT! =(ALERTD!-PBEFFECT(SSTBY(s)))/(PBSPEED!(SSTBY(s))-TSPEED!)

 IF (TSPEED!>PBSPEED!(SSTBY(s))) AND (ALERTD!>PBEFFECT(SSTBY(s))) THEN
    EXIT IF     'no intercept possible

   ELSE                                     'possible intercept
      TPOSS!(s) = TALERT! + TINTERCEPT!
      IF CMAX = 0 THEN CMAX = s
      IF TPOSS!(CMAX) >= TPOSS!(s) THEN
     STBYINCPT = s
     CMAX = s
      END IF
 END IF

 NEXT s

 IF  (TSAFE! > TPOSS!(CMAX)) AND (STBYINCPT > 0) THEN 'certain incpt
      TPT = 0
      STBYINCPTTOT = STBYINCPTTOT + 1
 SOPHOURS!(SSTBY(STBYINCPT))=SOPHOURS!(SSTBY(STBYINCPT)) + TPOSS!(CMAX)
      EXIT SUB
 END IF
```

```
END IF

END SUB


SUB REPLACESHIP(s,PLAT(),PLON(),PBHOME(),PBSPEED!())

  x = SHIPLAT(SDEP(s))
  y = SHIPLON(SDEP(s))

  TBLOCKSO!(SDEP(s)) = 10 * BLOCKSIZE!/(PBSPEED!(SDEP(s))/2)

  SOPHOURS!(SDEP(s)) = SOPHOURS!(SDEP(s)) + 2 'HOURS TO RETURN TO PORT

FOR m = 1 TO (NUMSSTBY + NUMSMAINT)

  OLD = SDEP(s)

IF NUMSMAINT > 0 THEN

    SDEP(s) = SMAINT(1)
  IF NUMSMAINT > 1.5 THEN
    FOR n = 1 TO (NUMSMAINT-1)
     SMAINT(n) = SMAINT(n+1)
    Next n
  END IF

  IF NUMSSTBY > 0 THEN

   SMAINT(NUMSMAINT) = SSTBY(1)

   IF NUMSSTBY > 1.5 THEN
    FOR n = 1 TO (NUMSSTBY -1)
     SSTBY(n) = SSTBY(n+1)
    NEXT n
   END IF

   SSTBY(NUMSSTBY) = OLD

  ELSE
   SMAINT(NUMSMAINT) = OLD
  END IF

ELSE
 SDEP(s) = OLD
END IF

                    'return deployed ship to port

SHIPLON(SSTBY(NUMSSTBY)) = PLON(PBHOME(SSTBY(NUMSSTBY)))
SHIPLAT(SSTBY(NUMSSTBY)) = PLAT(PBHOME(SSTBY(NUMSSTBY)))
SHIPLON(SMAINT(NUMSMAINT)) = PLON(PBHOME(SMAINT(NUMSMAINT)))
```

```
SHIPLAT(SMAINT(NUMSMAINT)) = PLAT(PBHOME(SMAINT(NUMSMAINT)))

  SHIPLAT(SDEP(s)) = x                    'replace ship on station
  SHIPLON(SDEP(s)) = y

  IF (SOPHOURS!(SDEP(s)) + 12) < (0.05 * ELT) THEN
   SOPHOURS!(SDEP(s)) = SOPHOURS!(SDEP(s)) + 2   'HOURS TO DEPLOY
   EXIT SUB
  END IF

NEXT m

SREST(SDEP(s)) = 1

END SUB


SUB REPLACEACFT(s,ALAT(),ALON(),AHOME$())

  x = ACFTLAT(ADEP(s))
  y = ACFTLON(ADEP(s))

    AOPHOURS!(ADEP(s)) = AOPHOURS!(ADEP(s)) + 1 'HOURS TO RTN TO BASE

FOR m = 1 TO NUMAMAINT

IF NUMAMAINT > 0 THEN          'prevents picking a non-existent acft

  OLD = ADEP(s)

  ADEP(s) = AMAINT(1)

IF NUMAMAINT > 1.5 THEN
  FOR n = 1 TO (NUMAMAINT-1)
   AMAINT(n) = AMAINT(n+1)
  Next n
END IF

  AMAINT(NUMAMAINT) = OLD
                              'return aircraft to base for maint
ACFTLAT(AMAINT(NUMAMAINT))=ALAT(ascii(AHOME$(AMAINT(NUMAMAINT)))-96)
ACFTLON(AMAINT(NUMAMAINT))=ALON(ascii(AHOME$(AMAINT(NUMAMAINT)))-96)

  ACFTLAT(ADEP(s)) = x                    'replace ACFT on station
  ACFTLON(ADEP(s)) = y
END IF


  IF (AOPHOURS!(ADEP(s)) + 4) < (0.05 * ELT) THEN
    AOPHOURS!(ADEP(s)) = AOPHOURS!(ADEP(s)) + 1   'HOURS TO DEPLOY
   EXIT SUB
  END IF

NEXT m
```

```
AREST(ADEP(s)) = 1

  END SUB


SUB RADARMATRIX(RLAT(),RLON(),PBRADRNG!(),ARADRNG!(),RRADRNG!(),e,ID$)
  'RADAR coverage matrix


SELECT CASE ID$

   CASE "s"          ' For ships

    For i = 1 to NUMSDEP
      x = SHIPLAT(SDEP(i))
      y = SHIPLON(SDEP(i))

      RNG! = PBRADRNG!(SDEP(i))/BLOCKSIZE!
       XMIN = x-RNG!
       XMAX = x+RNG!
       YMIN = y-RNG!
       YMAX = y+RNG!
      For j = XMIN to XMAX
       For k = YMIN to YMAX

       DISTANCE! =  (SQR(((j-x)^2) + ((k-y)^2)))

       IF DISTANCE! < RNG! THEN
         RADAR$(j,k) =  RADAR$(j,k) + "R"

       End If

       Next k
     Next j
   Next i

   CASE "a"    ' For aircraft

    For i = 1 to NUMADEP
      x = ACFTLAT(ADEP(i))
      y = ACFTLON(ADEP(i))

      RNG! = ARADRNG!(ADEP(i))/BLOCKSIZE!
       XMIN = x-RNG!
       XMAX = x+RNG!
       YMIN = y-RNG!
       YMAX = y+RNG!
      For j = XMIN to XMAX
       For k = YMIN to YMAX

       DISTANCE! =  (SQR(((j-x)^2) + ((k-y)^2)))
```

```
        IF DISTANCE! < RNG! THEN
          RADAR$(j,k) = RADAR$(j,k) + "R"
        End If

        Next k
      Next j
    Next i

    CASE "r"     ' For RADARs

     For i = 1 to e
       x = RLAT(i)
       y = RLON(i)

       RNG! = RRADRNG!(i)/BLOCKSIZE!
        XMIN = x-RNG!
        XMAX = x+RNG!
        YMIN = y-RNG!
        YMAX = y+RNG!
       For j = XMIN to XMAX
        For k = YMIN to YMAX

        DISTANCE! =  (SQR(((j-x)^2) + ((k-y)^2)))

        IF DISTANCE! < RNG! THEN
          RADAR$(j,k) = RADAR$(j,k) + "R"
        End If

        Next k
      Next j
     Next i

     CASE ELSE
       EXIT SELECT

    END SELECT

END SUB


SUB MIE

      PRINT "TOTAL INBOUND SMUGGLERS:  "; INTOT
      PRINT "TOTAL OUTBOUND SMUGGLERS: "; OUTTOT
      PRINT "TOTAL ILLEGAL FISHERMEN:  "; FISHTOT
      PRINT "TOTAL INBOUND SMUGGLERS INTERCEPTED:  ";  ININCPTTOT
      PRINT "TOTAL OUTBOUND SMUGGLERS INTERCEPTED: ";  OUTINCPTTOT
      PRINT "TOTAL ILLEGAL FISHERMEN INTERCEPTED:  ";  FISHINCPTTOT
      PRINT "TOTAL INTERCEPTS BY PATROLLING BOATS: ";  PATINCPTTOT
      PRINT "TOTAL INTERCEPTS BY STANDBY BOATS:    ";  STBYINCPTTOT

PRINT "MEASURE OF EFFECTIVENESS AGAINST INCOMING SMUGGLERS: ",MIEISMUG!
PRINT "MEASURE OF EFFECTIVENESS AGAINST OUTBOUND SMUGGLERS: ",MIEOSMUG!
PRINT "MEASURE OF EFFECTIVENESS AGAINST ILLEGAL FISHERMEN : ",MIEFISH!
```

```
END SUB

SUB COST

DIM PBACOST!(SZ),PBOCOST!(SZ),PBMCOST!(SZ),PBLIFE(SZ)
DIM AACOST!(AZ),AOCOST!(AZ),AMCOST!(AZ),ALIFE(AZ)
DIM RACOST!(RZ),ROCOST!(RZ),RMCOST!(RZ),RLIFE(RZ)

    'Order of Battle input

 OPEN NOBF$ for input as #10
p = 0
WHILE not EOF(10)
p = p + 1
  INPUT #10, SCNAM$,PBSPEED!,PBRADRNG!,PBENDUR,PBEFFECT,_
    PBACOST!(p),PBOCOST!(p),PBMCOST!(p),PBLIFE(p),PBHOME

    'FACTOR IN OPERATING HOURS

    PBOCOST!(p) = SOPHOURS!(p)*PBOCOST!(p)
    PBMCOST!(p) = SOPHOURS!(p)*PBMCOST!(p)

    'total costs
    SOCOSTTOT!=SOCOSTTOT! + PBOCOST!(p)
    SMCOSTTOT!=SMCOSTTOT! + PBMCOST!(p)

LOOP
 CLOSE #10

 OPEN AOBF$ for input as #10
a = 0

WHILE not EOF(10)
a = a + 1

    INPUT #10, ACNAM$,ASPEED!,ARADRNG!,AENDUR,_
     AACOST!(a),AOCOST!(a),AMCOST!(a),ALIFE(a),AHOME$

    'factor in OPHOURS

    AOCOST!(a) = AOPHOURS!(a)*AOCOST!(a)
    AMCOST!(a) = AOPHOURS!(a)*AMCOST!(a)

    'total costs
    AOCOSTTOT!=AOCOSTTOT! + AOCOST!(a)
    AMCOSTTOT!=AMCOSTTOT! + AMCOST!(a)
LOOP
 CLOSE #10

 OPEN EOBF$ for input as #10
e = 0
```

```
WHILE not EOF(10)
e = e + 1

   INPUT #10, RNAM$,RRADRNG!,_
    RACOST!(e),ROCOST!(e),RMCOST!(e),RLIFE(e),RHOME$

    'total costs
    ROCOSTTOT!=ROCOSTTOT! + ROCOST!(e)
    RMCOSTTOT!=RMCOSTTOT! + RMCOST!(e)

    'CORRECT FOR PERIOD RUN
    ROCOSTTOT! = ROCOSTTOT! * (DAYS/365.24)
    RMCOSTTOT! = RMCOSTTOT! * (DAYS/365.24)

LOOP
 CLOSE #10

   'Retrieve threat data

 OPEN INTHR$ for input as #1
  Input #1, IND!,INN!,INC!,INS!,INA!
 Close #1

  OPEN OUTTHR$ for input as #1
  Input #1, OUTD!,OUTN!,OUTC!,OUTS!,OUTA!
 Close #1

 OPEN FISHTHR$ for input as #1
  Input #1, FISHD!,FISHN!,FDST!,FISHC!,FISHS!,FISHA!
 Close #1

 ' COST CALCULATIONS


INTOTSAVE! = INC! * ININCPTTOT
OUTTOTSAVE! = OUTC! * OUTINCPTTOT
FISHTOTSAVE! = FISHC! * FISHINCPTTOT
OMCOSTTOT!= SOCOSTTOT! + SMCOSTTOT! + AOCOSTTOT! + AMCOSTTOT! + _
          ROCOSTTOT! + RMCOSTTOT!

 Print "Money saved through intercepting for "; DAYS;" days:"
 Print "Inbound Smugglers: $";INTOTSAVE!
 Print "Outbound Smugglers:$";OUTTOTSAVE!
 Print "Illegal Fishing:   $";FISHTOTSAVE!
 Print
 Print "Costs of operations and maintenance for "; DAYS;" days:"
 Print "Ship operating costs: $";SOCOSTTOT!
 Print "Ship maintenance costs $";SMCOSTTOT!
 Print "Aircraft operating costs: $";AOCOSTTOT!
 Print "Aircraft maintenance costs $";AMCOSTTOT!
 Print "Radar operating costs: $";ROCOSTTOT!
 Print "Radar maintenance costs $";RMCOSTTOT!
 Print
 Print "Total operating and maintenance costs for "; DAYS;" days"
```

```
 Print
 Print "$";OMCOSTTOT!

 While not instat : wend
 k$ = INKEY$
 cls

 Print "Annual savings at this rate: "
 Print "Inbound Smugglers: $";INTOTSAVE!*365.24/DAYS
 Print "Outbound Smugglers:$";OUTTOTSAVE!*365.24/DAYS
 Print "Illegal Fishing:   $";FISHTOTSAVE!*365.24/DAYS
 Print
 Print "Annual operating and maintenance costs at this rate: "
 Print

 YROMCOSTTOT!= OMCOSTTOT!*356.24/DAYS
 Print "$"; YROMCOSTTOT!
DO

 INPUT "Input interest rate (percent): "; INTEREST!
 INPUT "Input inflation rate (percent): "; INFLATION!
 Print

      'calculating the inflation free rate
      IFR!= ((1+INTEREST!/100)/(1+INFLATION!/100))-1

   'calculating annual equivalent life cycle costs of acquisition

 For i = 1 to p
SALIFECOST!=PBACOST!(p)*_
          ((IFR!*(1+IFR!)^PBLIFE(p))/(((1+IFR!)^PBLIFE(p))-1))
TOTSALIFECOST!= TOTSALIFECOST! + SALIFECOST!
 NEXT i

 For i = 1 to a
AALIFECOST!=AACOST!(a)*_
          ((IFR!*(1+IFR!)^ALIFE(a))/(((1+IFR!)^ALIFE(a))-1))
TOTAALIFECOST!= TOTAALIFECOST! + AALIFECOST!
 NEXT i

 For i = 1 to e
RALIFECOST!=RACOST!(e)*_
          ((IFR!*(1+IFR!)^RLIFE(e))/(((1+IFR!)^RLIFE(e))-1))
TOTRALIFECOST!= TOTRALIFECOST! + RALIFECOST!
 NEXT i

 COSTTOT!=YROMCOSTTOT!+TOTSALIFECOST!+TOTAALIFECOST!+TOTRALIFECOST!
 SAVETOT! = (INTOTSAVE! + OUTTOTSAVE! + FISHTOTSAVE!) * (365.24/DAYS)
 NETTOT! = COSTTOT! - SAVETOT!

Print
Print "Annual equivalent cost of Patrol Boat Acquisition: $";_
      TOTSALIFECOST!
Print "Annual equivalent cost of Aircraft Acquisition: $";_
```

```
       TOTAALIFECOST!
Print "Annual equivalent cost of Radar Acquisition: $";_
       TOTRALIFECOST!
Print
Print "Total annual equivalent costs:                          $";_
       COSTTOT!
Print "Total annual equivalent savings:                        $";_
       SAVETOT!
Print "Net annual equivalent cost of coastal defense system: $";_
       NETTOT!
Print

 Input "Do you want to change the interest or inflation rates"; logi$
     lo$= lcase$(left$(logi$,1))
   If lo$ <> "y" then
    EXIT SUB
   End if

LOOP

END SUB

GERROR:

IF ERR = 53 THEN
  PRINT "File not found"
  Resume
Else
  PRINT "Error "; ERR; " in line "; ERL ; " address "; ERADR
  END
end if
return

gexit:

INPUT "DO YOU REALLY WANT TO EXIT"; LOGI$
    if lcase$(left$(logi$,1)) = "y"  THEN
    COLOR 7,0,0
    CLS
    STOP
    end if
RETURN

end
```

Appendix B: Definitions of Variables Used

a: Total number of aircraft in the OOB.

AACOST!(): Acquisition cost of aircraft ($).

AACOSTTOT!: Total cost of aircraft acquisition.

AALIFECOST!: Aircraft acquisition annual equivalent life cycle cost.

ACFTLAT(): Latitude of deployed or standby aircraft.

ACFTLON(): Longitude of deployed or standby aircraft.

ACLASS: File name of available aircraft types.

ACLASSCASE: Value of Aircraft Type menu selection for SELECT CASE routine.

ACNAM$(): Aircraft type.

ADEP(): Sequential number assigned to aircraft deployed.

ADEPAGE(): Duration of aircraft's current deployment.

AENDUR(): Endurance (hours per patrol of aircraft).

AFZ: Dimension for arrays of airfield locations.

AHOME$(): Home base of aircraft.

ALAT(): Latitude of airfield.

ALERTD!: Target alertment distance.

ALIFE(): Life time of aircraft in years.

ALON(): Longitude of airfield.

AMAINT(): Sequential number assigned to aircraft in maintenance.

AMCOST!(): Maintenance cost of aircraft ($/year).

AMCOSTTOT!: Total cost of aircraft maintenance.

AOB: Name of the Maritime Air order of battle generation procedure.

AOBF$: Name of Maritime Air Patrol Order of Battle File (=CTYCODE$.AOB).

AOCOST!(): Operating cost of aircraft ($/year).

AOCOSTTOT!: Total cost of aircraft operation.

AOPHOURS!(): Aircraft operating hours.

APATROL!: Percent of aircraft on patrol.

APLON(): Longitude of air patrol box (uses latitude as its argument).

ARADRNG!(): Radar range of aircraft.

AREST(): Identifies if a aircraft is overworked

ASPEED!(): Max patrol speed of aircraft.

ATYPE: Name of the aircraft type procedure.

AZ: Dimension for arrays of maritime air order of battle.

BACKA(): The direction of aircraft movement (0 counts up, 1 counts down).

BACKS(): The direction of ship movement (0 counts up, 1 counts down).

BCOL$: Lowercase value of individual characters in LAYOUT$ for SELECT CASE.

BEACHCASE$: Value of each two letter LAYOUT$() combo for SELECT CASE routine.

BLOCKSIZE!: Size of each block in LAYOUT$ array in NM.

BOXPERACFT: Number of patrol boxes assigned per aircraft.

BOXPERSHIP: Number of patrol boxes assigned per ship.

CHECK1$: First letter of BEACHCASE$ if last letter is "w".

CHECK2$: First letter of BEACHCASE$ if last letter is "l".

CLON(): Longitude of coastline (uses latitude as its argument).

CMAP$(): Line by line input variable for physical layout input.

CMAX: Deployed number of closest patrol craft to target.

CNAME$: Name of coastal area.

COAST$(): Two value array, each representing successive values of LAYOUT$().

COASTC: Name of the coastal configuration procedure.

COASTCASE: Value of coastal configuration menu selection for SELECT CASE.

COASTD: Distance from target to coast.

COLD$(): Array of physical layout names on file.

COMPLF!: complementary random pair calculations for illegal fishermen.

COMPLI!: complementary random pair calculations for inbound smugglers.

COMPLO!: complementary random pair calculations for outbound smugglers.

COSTTOT!: Total annual equivalent costs for coastal defense system.

CTYCODE$: Two letter abbreviation for country name.

d: Half day counter in simulation routine.

DAPATROL!: Aircraft patrol distance from coast.

DAYC!: Counts days of simulation (d/2).

DAYS: The number of days to run the model (user input).

DEPF$: Name of Deployment Data File (=CTYCODE$.DEP).

DEPLOY: Name of the deployment data procedure.

DISCST: Name of the subroutine to redraw the physical layout on screen.

DISTANCE!: Distance of radar coverage (in blocks).

DN: Input variable of number of the ship or aircraft to be deleted.

DSPATROL!: Ship patrol distance from coast.

e: Total number of shore based radars in the OOB.

EDGE: Defines which edge of the screen has water.

EEZ: Distance from land that a country claims as its Exclusive Economic Zone.

EEZ$: Name of claimed EEZ file (EEZ.CTYCODE$).

EEZD: Distance from target to EEZ.

EEZEDGE(): Longitude of edge of EEZ (uses latitude as its argument).

ELT: Elapsed time in the simulation (in increments).

EOB: Radar (Electronic) order of battle generation procedure.

EOBF$: Name of Radar (Electronic) Order of Battle File (=CTYCODE$.EOB).

FDST!: Average distance of fishing banks from coast.

FISHA!: Alertment distance for illegal fishermen.

FISHARR: Arrival rate for illegal fishermen.

FISHC!: Average illegal fishing boat cost to the economy.

FISHD!: Average number of illegal fishermen during the day.

FISHGRID$(): Array to track illegal fishermen to compare with RADAR$().

FISHINCPTTOT: The total number of illegal fishermen intercepted.

FISHLAT(): Latitude of illegal fisherman.

FISHLON(): Longitude of illegal fisherman.

FISHN!: Average number of illegal fishermen during the night.

FISHS!: Average fishing boat max speed.

FISHTHR$: Name of Illegal Fishing Threats Data File (=FISHTHR.CTYCODE$).

FISHTHREAT: Current number of illegal fishermen in simulation.

FISHTOT: The total number of illegal fishermen during the simulation.

FISHTOTSAVE!: Total savings of intercepting illegal fishermen.

GERROR: The name of the error handling subroutine.

GEXIT: The name of the subroutine to exit on the <F1> key.

H: Input variable for Home base of selected boats, aircraft, or radars.

ID$: SELECT CASE value to determine if ship, airborne, or shore radar.

IFR!: Inflation free interest rate.

INA!: Alertment distance for inbound smugglers.

INARR: Arrival rate for inbound smugglers.

INC!: Average inbound smuggler cost to the economy.

IND!: Average number of inbound smugglers during the day.

INFLATION!: Inflation used to calculate equivalent costs.

INGRID$(): Array to track inbound smugglers to compare with RADAR$().

ININCPTTOT: The total number of inbound smugglers intercepted.

INLAT(): Latitude of inbound smuggler.

INLON(): Longitude of inbound smuggler.

INN!: Average number of inbound smuggler during the night.

INS!: Average inbound smuggler speed.

INSTAT: Records when any key is pushed on the keyboard.

INTEREST!: Interest (discount rate) used to calculate equivalent costs.

INTHR$: Name of Inbound Threats Data File (=INTHR.CTYCODE$).

INTHREAT: Current number of inbound smugglers in simulation.

INTOT: The total number of inbound smugglers during the simulation.

INTOTSAVE!: Total savings of intercepting inbound smugglers.

ISEPD!: Target to patrol boat separation at time of detection (NM).

K$: Holds value of INSTAT to prevent interference with rest of program.

L: Input variable for line number of selected boats, aircraft, or radars.

LAYOUT(): The physical characteristics of the coast.

LESSTHREAT: Subroutine to decrement threat contacts.

LO$: Input variable for yes or no questions.

LOGI$: Input variable for yes or no questions.

MAINCASE: Value of main menu selection for SELECT CASE routine.

MAINMENU: Name of the main menu procedure.

MIE: Procedure to calculate the measure of intercept effectiveness.

MIEFISH: Measure of intercept effectiveness for illegal fishermen.

MIEISMUG: Measure of intercept effectiveness for inbound smugglers.

MIEOSMUG: Measure of intercept effectiveness for outbound smugglers.

NETTOT!: COSTTOT! - SAVETOT!.

NOB: Coast Guard (Naval) order of battle generation procedure.

NOBF$: Name of Coast Guard (Naval) Order of Battle File (=CTYCODE$.NOB).

NUMADEP: Number of aircraft deployed at any time.

NUMAMAINT: Number of aircraft in maintenance.

NUMASTBY: Number of aircraft on 30 minute standby at any time.

NUMSDEP: Number of ships deployed at any time.

NUMSMAINT: Number of ships in maintenance.

NUMSSTBY: Number of ships on 30 minute standby at any time.

OMCOSTTOT!: Total costs for operation and maintenance during simulation.

OOB: Name of the order of battle generation procedure.

OOBCASE: Value of Order of Battle menu selection for SELECT CASE routine.

OUTA!: Alertment distance for outbound smugglers.

OUTARR: Arrival rate for outbound smugglers.

OUTC!: Average outbound smuggler cost to the economy.

OUTD!: Average number of outbound smugglers during the day.

OUTGRID$(): Array to track outbound smugglers to compare with RADAR$().

OUTINCPTTOT: The total number of outbound smugglers intercepted.

OUTLAT(): Latitude of outbound smuggler.

OUTLON(): Longitude of outbound smuggler.

OUTN!: Average number of outbound smuggler during the night.

OUTS!: Average outbound smuggler speed.

OUTTHR$: Name of Outbound Threats Data File (=OUTTHR.CTYCODE$).

OUTTHREAT: Current number of outbound smugglers in simulation.

OUTTOT: The total number of outbound smugglers during the simulation.

OUTTOTSAVE!: Total savings of intercepting outbound smugglers.

p: Total number of ships in the OOB.

PATINCPTTOT: The total number of intercepts by boats on patrol.

PBACOST!(): Acquisition cost of patrol boat ($).

PBEFFECT(): Effectiveness of patrol boat.

PBENDUR(): Endurance (hours at sea) of patrol boat.

PBHOME(): Home base of patrol boat.

PBLIFE(): Life time of patrol boat in years.

PBMCOST!(): Maintenance cost of patrol boat ($/year).

PBOCOST!(): Operating cost of patrol boat ($/year).

PBRADRNG!(): Radar range of patrol boat.

PBSPEED!(): Max sustained speed of patrol boat.

PLAT(): Latitude of port.

PLAYOUT: Name of the physical layout procedure.

PLON(): Longitude of port.

PZ: Dimension for arrays of port locations.

Q: Input variable for Quantity of selected boats, aircraft, or radars.

RACOST!(): Acquisition cost of shore based radar ($).

RACOSTTOT!: Total cost of radar acquisition.

RADAR$(): Array of total radar coverage ("R" represents coverage).

RADARMATRIX: Name of the radar array generation procedure.

RALIFECOST!: Radar acquisition annual equivalent life cycle cost.

RANDOMMOVE: Random direction for illegal fishermen movement SELECT CASE.

RCLASS: File name of available radar types.

RCLASSCASE: Value of Radar Type menu selection for SELECT CASE routine.

REPLACEACFT: Procedure to replace a deployed aircraft.

REPLACESHIP: Procedure to replace deployed ship with one from standby.

RHOME$(): Home base of radar.

RLAT(): Latitude of shore based radar.

RLIFE(): Life time of shore based radar in years.

RLON(): Longitude of shore based radar.

RMCOST!(): Maintenance cost of shore based radar ($/year).

RMCOSTTOT!: Total cost of radar maintenance.

RNAM$(): Radar name.

RNG!: Range calculation variable (in blocks).

ROCOST!(): Operating cost of shore based radar ($/year).

ROCOSTTOT!: Total cost of radar operation.

RRADRNG!(): Radar range of shore based radar.

RZ: Dimension for arrays of shore based radar order of battle.

SACOSTTOT!: Total cost of ship acquisition.

SAFED!: The target's distance from shore or EEZ limit.

SALIFECOST!: Ship acquisition annual equivalent life cycle cost.

SAVETOT!: Total savings each year from threat interception.

SC!(): Array of the scales for physical layouts on file.

SCALE!: The scale of the coast being used. (ie. 1: SCALE!).

SCLASS: File name of available ship classes.

SCLASS: Name of the ship class procedure.

SCLASSCASE: Value of Ship Class menu selection for SELECT CASE routine.

SCNAM$(): Ship class name.

SCREENINPT: Name of the subroutine to input physical layout data.

SDEP(): Sequential number assigned to ships deployed.

SDEPAGE(): Duration of ship's current deployment.

SHIPLAT(): Latitude of deployed or standby ship.

SHIPLON(): Longitude of deployed or standby ship.

SIMULATE: Name of the main simulation procedure.

SMAINT(): Sequential number assigned to ships in maintenance.

SMCOSTTOT!: Total cost of ship maintenance.

SOCOSTTOT!: Total cost of ship operation.

SOPHOURS!(): Ship operating hours.

SPATROL!: Percent of ships on patrol.

SPLON(): Longitude of ship patrol box (uses latitude as its argument).

SRADAR: Name of the shore based radar type procedure.

SREST(): Identifies if a ship is overworked

SSTBY(): Sequential number assigned to ships on 30 minute standby.

STBYINCPT: Deployed number of intercepting standby patrol craft.

STBYINCPTTOT: The total number of intercepts by boats on standby.

SZ: Dimension for arrays of coast guard order of battle.

t: Time in simulation routine (each increment 0.1 hour (6 minutes)).

TALERT!: The time to reach the alert radius.

TBLOCKA!(): Cumulative time that aircraft has moved.

TBLOCKAO!(): Time for the aircraft to move one block.

TBLOCKFISH!(): Cumulative time that illegal fisherman has moved.

TBLOCKFISHO!(): Time for the illegal fisherman to move one block.

TBLOCKIN!(): Cumulative time that inbound smuggler has moved.

TBLOCKINO!(): Time for the inbound smuggler to move one block.

TBLOCKOUT!(): Cumulative time that outbound smuggler has moved.

TBLOCKOUTO!(): Time for the outbound smuggler to move one block.

TBLOCKS!(): Cumulative time that ship has moved.

TBLOCKSO!(): Time for the ship to move one block.

TFISH!: Interarrival time for next illegal fishermen.

TFISHCHECK: Cumulative counter of interarrival time for illegal fishermen.

THREAT: Name of the threat data procedure.

THREATCASE: Value of Threat menu selection for SELECT CASE routine.

TIN!: Interarrival time for next inbound smuggler.

TINCHECK!: Cumulative counter of interarrival time for inbound smugglers.

TINTERCEPT!: Time to intercept after alertment.

TOTAALIFECOST!: Total aircraft acquisition annual equivalent life cycle cost.

TOTRALIFECOST!: Total radar acquisition annual equivalent life cycle cost.

TOTSALIFECOST!: Total ship acquisition annual equivalent life cycle cost.

TOUT!: Interarrival time for next outbound smuggler.

TOUTCHECK: Cumulative counter of interarrival time for outbound smugglers.

TPOSS!(): Array to track the possible intercept ships to choose the closest.

TPT: Value to determine if intercept was successful.

TPT1: Deployed number of intercepting ship.

TSAFE!: The time for the target to reach safety.

TSPEED!: Target speed (FISHS! or INS! or OUTS!).

x: For-next loop variable used to represent latitude.

XMAX: Corner coordinate of square representing radar coverage.

XMIN: Corner coordinate of square representing radar coverage.

y: For-next loop variable used to represent longitude.

YMAX: Corner coordinate of square representing radar coverage.

YMIN: Corner coordinate of square representing radar coverage.

YROMCOSTTOT!: Total annual equivalent costs for operation and maintenance.

! Single Precision real variable

$ String variable

() Array

All undesignated variables are integers

Latitude and Longitude are used loosely to give a picture of the x-y axis.