NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPUTER GRAPHICS ENHANCEMENT FOR COMEL

by

Martin R. Conrad, Jr.

March, 1984

Thesis Advisor: G. R. Porter

Approved for public release, distribution unlimited

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Computer Graphics Enhancement for COMEL

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis
March, 1984

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Martin R. Conrad, Jr.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93943

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93943

12. REPORT DATE
March, 1984

13. NUMBER OF PAGES
111

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
DI - 3000
Computer graphics enhancement
COMEL
hexagonal matrix
computer war game
Communications electronics war game
Ramtek

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

COMEL is a communications-oriented war game developed by the Joint Telecommunications Staff Officer's Course at Keesler AFB. The war game was automated by previous thesis students to run on a VAX/VMS computer. This thesis further modified COMEL to eliminate the manual gameboard and totally automate the game using Ramtek graphic display devices.

The game has two portions, an Acquisition Phase and an Operations Phase. In the Acquisition Phase, players budget for research and development, manufacturing, purchase, and operations and maintenance of communications and electronic equipment for a Joint Task Force (JTF). In the Operations Phase, players allocate the available communications and electronic equipment to units, physical locations, or special missions and then direct the employment of the units and equipment in a war game.

The programs are written in structured FORTRAN-77, with extensive comments and external documentation. The graphics routines were written using the DI-3000 graphics system.

Approved for public release, distribution unlimited

Computer Graphics Enhancement for COMEL

by
Martin R. Conrad, Jr.
Captain, United States Army
B.S., United States Military Academy, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(Command, Control, and Communications)

from the

NAVAL POSTGRADUATE SCHOOL
March 1984

ABSTRACT

COMEL is a communications-oriented war game developed by the Joint Telecommunications Staff Officer's Course at Keesler AFB. The war game was automated by previous thesis students to run on a VAX/VMS computer. This thesis further modified COMEL to eliminate the manual gameboard and totally automate the game using Ramtek graphic display devices.

The game has two portions, an Acquisition Phase and an Operations Phase. In the Acquisition Phase, players budget for research and development, manufacturing, purchase, and operations and maintenance of communications and electronic equipment for a Joint Task Force (JTF). In the Operations Phase, players allocate the available communications and electronic equipment to units, physical locations, or special missions and then direct the employment of the units and equipment in a war game.

The programs are written in structured FORTRAN-77, with extensive comments and external documentation. The graphics routines were written using the DI-3000 graphics system.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

Communications Electronics War (COMEL) was developed as a manual war game in 1982 by students at the Joint Telecommunications Systems Staff Officers Course (TSSOC) at Keesler Air Force Base, Mississippi. The goals of the game were to "integrate selected objectives of TSSOC into a single exercise" and to give an understanding of "the complexities of joint communications systems acquisition and planning." [Ref. 1]

The basic concept of the game was developed by Major James L. Parrine and other TSSOC faculty members. Their concept called for a game based on the assumption that "operational forces are only as effective as the comm supporting them." [Ref. 1] Their initial concept was primarily concerned with acquisition. Each team was to begin the game year with given operational communications and electronics systems. They are also given a "shopping list" of more sophisticated systems to upgrade their combat capabilities. The teams would have research, acquisition, and intelligence decisions to make with limited resources and a time constraint. Subsequent turns would represent budget years. The time limit for each turn is decided by the umpire. After each turn there is a probability of going

to war so the players have to plan their strategies to acquire systems as rapidly as possible without exceeding their budgets. Random events would be entered into the game by rolling dice and via "Gimme a Break" and "Awshucks" cards. The initial concept called for the development of an objective evaluation criteria to determine the game winner based on acquisition strategies only.

The concept was further refined to include a combat phase with faculty members acting as controllers and evaluators. The first turn of the combat phase involved allocating the fielded equipment (allocation phase) to combat units in the field in an attempt to optimize combat unit equipment with communications connectivity. Preliminary design goals were defined and given to their students to implement as a major project of their course.

The goal of the thesis written by Rowe and Allgood was to develop an automated computer version of COMEL to incorporate most of the features of the manual version and to allow later expansion of the game [Ref 1]. The computer version was designed in two portions. Allgood designed the control modules and the Acquisition Phase modules; Rowe designed the Operations Phase modules. The two portions were then combined to form a single game. Rowe and Allgood succeeded in fully automating COMEL with the exception of the gameboards that are used in the operations phase of the game.

The computer version is resident in the Wargaming, Analysis, and Research Laboratory (WAR Lab) at the Naval Postgraduate School (NPS) and can be modified for use at the TSSOC or at any other computer facility with a compatible system. COMEL is written in FORTRAN-77 and is currently running on a VAX 11-780 with the VMS operating system. In addition to the basic thesis, the Rowe/Allgood thesis also contains a detailed users' manual and maintenance manual.

B. THESIS GOAL

The goal of this thesis was to provide graphics enhancements to the operations phase of the Rowe/Allgood thesis [Ref 1] that would result in a fully automated COMEL game that will enable the participants to concentrate their efforts on the execution of the wargame. Manipulating game pieces on a manual gameboard proved to be a tedious task that detracted from the primary purpose of the game and was a possible source of errors.

The graphics displays were achieved through the use of the DI-3000 graphics software and the Ramtek 9460 graphics display system. The DI-3000 system allows graphics routines to be called from FORTRAN programs, permitting the addition of a significant amount of graphics to the Rowe/Allgood thesis with minor changes to their program.

The DI-3000 graphics software system is a device-independent computer graphics programming system that was

designed by Precision Visuals, Inc., and enhanced by Lawrence Livermore Laboratories. Device-independence implies that the graphics output will appear to be similar when displayed on different graphics display devices.

A modified users' manual and a modified maintenance manual have been provided as appendices to this thesis and include the changes made to reference 1 as a result of the graphics enhancements.

C. ORGANIZATION

The remainder of this thesis is partitioned into the following chapters:

Chapter II - Comel Overview

A comprehensive review of COMEL is provided in this chapter. The purpose and object of COMEL are discussed briefly. The acquisition phase and the operations phase are condensed and examined to provide some insight into the game.

Chapter III - COMEL Corrections

Software errors were discovered in the original version of COMEL and corrected. The corrections are discussed in this chapter.

Chapter IV - General Approach to Graphics

This chapter explains the approach taken to apply graphics routines to COMEL. The DI-3000 graphics system is discussed so the methods used to present the graphics are

understood better. The basic gameboard and unit displays are also explained so the theory behind the methods is understood. Finally, the methods that were used to generate the graphics display are explained.

Chapter V - Conclusions

The results of the thesis are discussed in this chapter.

Chapter VI - Recommendations

Several other enhancements are discussed in this chapter. As the original graphics design was developed other areas were discovered that could be better understood using graphics displays. Enhancements for the acquisition phase and the allocation phase are discussed as are further enhancements for the operations phase.

Appendix A - COMEL Users' Manual

This user's manual contains an abbreviated version of the original one created by Rowe and Allgood and contains detailed instructions on the use of the COMEL graphics enhancements.

Appendix B - COMEL Maintenance Manual

This manual is patterned after the Rowe/Allgood maintenance manual. It only covers the subroutines that were modified or created to allow the graphics game to be run.

I. COMEL OVERVIEW

COMEL is a joint communications-electronics war game played by a Red Team, a Green team, and an Umpire. It was developed primarily to train communications staff officers. The first version of the game was completely manual. The next version was the computer version that was created by Rowe and Allgood that still uses a manual gameboard. The third version, using graphic enhancements, will eliminate the manual gameboard and automate the unit movements.

The object of COMEL is to acquire electronics systems that can be allocated to units on the battlefield. The units, under the direction of the Joint Task Force Headquarters (JTFHQ) will then maneuver and attempt to take certain objectives on the battlefield. The JTFHQ's for the Red and Green Forces are controlled by the National Command Authority (NCA). The NCA for COMEL is the Umpire.

COMEL simulates the three phases of the system life cycle. The planning phase of the cycle is done prior to beginning the computer version of the game. The second phase, acquisition, is simulated in the acquisition phase of COMEL. The third phase, usage, is simulated in the operations phase of the game.

The game is divided into two phases, the acquisition phase and the operations phase; both can be played during

one session or over more than one session. The umpire can save the game at any point for resumption of the game at the next session. There are default files in the data base to provide both sides with a rudimentary set of electronic equipment for use during the acquisition phase. Likewise, there are default files that can be used if the umpire and players want to go directly to the operations phase of the game without playing the acquisition phase. Players can design games with different sets of electronic equipment, units, gameboard, and probabilities or use default files provided with the game.

A. ACQUISITION PHASE

The first phase, acquisition, includes all those steps necessary to design, test and evaluate, produce, and install the planned systems [Ref. 1]. It permits evaluation of different systems for possible acquisition and allocation to the combat forces in the field. Each game turn is considered to be a budget year.

While in the acquisition phase, the Planning, Programming, and Budgeting System (PPBS) cycle is utilized. The steps of the PPBS cycle that are used in COMEL are (1) deciding which systems to buy, (2) putting systems into advanced Research and Development (AR&D) or normal R&D, (3) advancing systems into Manufacturing for Deployment (M&D), (4) advancing systems into operations and maintenance

(O&M), and (5) scrapping systems when no longer needed. Not all equipment needs R&D; some equipment may proceed directly to M&D. These phases are explained in more detail in Reference 1.

While in the acquisition phase the equipment can also go through three phases. The first phase is the actual procurement of the equipment. The second phase is the O&M phase where O&M costs are incurred. The final phase involves scrapping the equipment if it proves to be too costly or is no longer needed.

When the acquisition phase of the game begins, a menu is presented that gives several options.

1. Shopping List

This option shows a shopping list of equipment and systems available for purchase. If the shopping list option is chosen, other options are displayed to provide general information, technical information, and/or cost information on a specific piece of equipment or system.

2. Systems Status

When this option is chosen, a submenu is displayed that makes available an updated systems summary, a list of systems deployed, systems in R&D, systems that have finished R&D and are ready for M&D, systems in M&D, and systems available to be bought.

3. Intelligence Reports

This option provides intelligence reports on the status of the opposing force's systems status. To get the intelligence a request for the intelligence must be sent to the umpire. The request costs money. The umpire has the option of giving the player no information, a little bit of information, or a great deal of information.

4. Budget Analysis/Request

The budget analysis/request displays a five-year cost projection and allows the budget request for the next year to be submitted. By using this option information can be obtained to decide whether to allocate resources to have equipment developed in real time or have it placed in advanced R&D so it can be available sooner. Requests for a budget for the next year are also submitted. The umpire has the option of allocating the amount that has been requested or allocating more or less than was requested. The program is designed so the umpire cannot give the players too large or too small a budget.

5. Procurement

The final option deals with the procurement of the equipment and systems. If this option is selected help can be obtained thru a decision analysis aid, systems can be bought, systems can be scrapped (if the remaining funds are running low), equipment and systems in R&D can be placed in accelerated R&D, the R&D effort can be expanded, O&M can be

reduced, intelligence can be purchased from the umpire or messages can be sent to the umpire, and systems can be placed in M&D.

When both sides have made made all of their decisions or run out of money or run out of time, the turn is complete. The umpire then makes the next move.

Another acquisition turn can then be played or war can begin at this point. If the decision is to continue the acquisition phase, the umpire decides how much money to allocate to the red and green players. At this point the umpire can also read messages that were sent by the players and review requests for intelligence.

If the umpire decides to terminate the acquisition phase of the game and begin the war, the game enters the allocation phase.

B. THE OPERATIONS PHASE

The operations phase of the game is played on a matrix gameboard consisting of sixty-six rows with thirty hexes per row (figure 1).

If only the operations phase of the game is played, the Red and Green teams can play the game with default forces, a default gameboard, and default equipment. The forces and equipment are listed below without any capabilities listed. A complete description of the units and equipment can be found in Reference 1.

The Green player has the following units available:

jtfhq
4th cbg
37 mech
6th tfw
31st tfw
64th atf
437 arbg
194 arbg
87 abn
6th mab

Default equipment available to the Green player consists of the following:

artac 1 (4)
artac 2
navtac 2 (3)
aftac 2 (2)
hftty 1 (12)
wbs 2 (6)
awacs (4)
shf-gt (6)
vlf
tritaceh
sat 3
gt-sat 3 (3)
abncp 2 (2)
spysat 1 (2)
atksat 2 (2)
atksat 1
ew 2 (5)
ew 3
ew 4a

The Red player has the following default units available:

jtfhq
24th tfw
86th tfw
7th cbg
21 atf
8th mab
297 arbg
81 arbg
8th abn
41 mech

The default equipment available to the Red player consists of:

artac 1
artac 2 (4)
navtac 1 (3)
aftac 1 (2)
hftty 2 (12)
wbs 1 (6)
shf-sat
shf-gt (3)
vlf
tritac
sat 1
gt-sat 1 (3)
singars
singargt (3)
abncp 1 (3)
awacs-eh (2)
awacs
spysat 2 (2)
atksat 1 (3)
ew 1 (3)
ew 3a
ew 5a

The operations portion of the game is actually divided into two distinct phases, the allocation phase and the actual operations or combat phase.

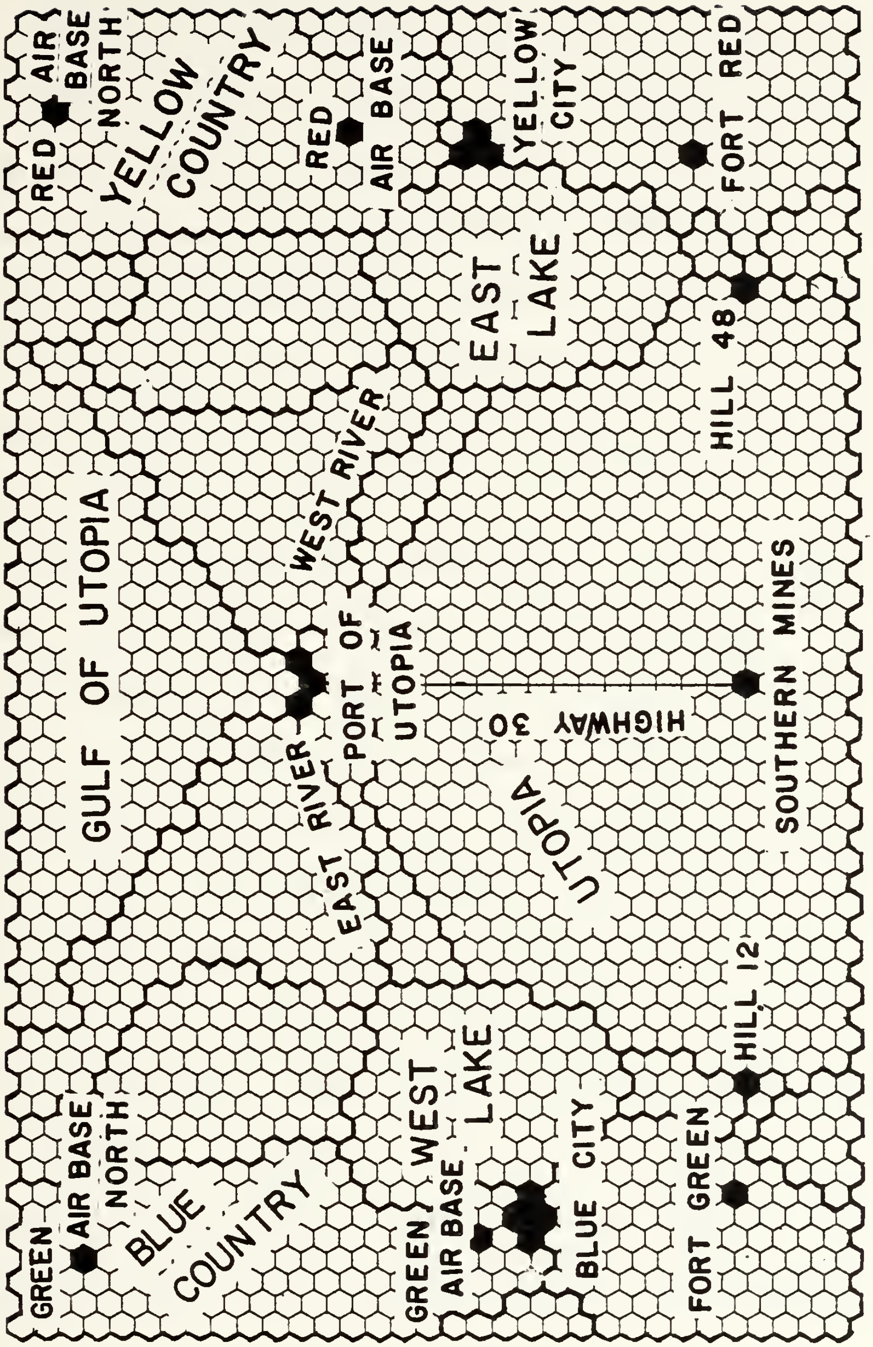


Figure 1. Basic Gameboard.

1. The Allocation Phase

During the allocation phase of the game, the red and green players allocate equipment and systems to their units. The only equipment and systems that are available to be allocated are those that completed the M&D cycle. Since the game turns in the operations phase of the game simulate one day while the game turns in the acquisition phase are a year long, the war will be over before equipment and systems in R&D or M&D can be fielded.

When allocating equipment, the players must understand the capabilities of the equipment. For units to be able to communicate with each other and with higher authority, their equipment must be compatible. The COMEL program checks regularly for connectivity between units based on the distance between the units, the type of equipment being used, the terrain features, and a random factor generated by the computer to simulate changing atmospheric effects. When the Joint Task Force Headquarters (JTFHQ) issues orders to units, the units must have connectivity with JTFHQ to receive the orders. If there is no connectivity, the units cannot be ordered to move or fight.

After the allocation phase has been completed, the operations phase is started.

2. The Operations Phase

During the operations phase of the game, the communications links are affected by range, connectivity, terrain, and other factors that simulate environmental conditions. A communications link may or may not work. Proper allocation of equipment will reduce "communications outages."

a. Terrain

Movement, combat, and communications also affected by types of terrain encountered.

(1) Open terrain. Open terrain has no effect on the communications system. Moving across open terrain costs a unit one movement point per hex. There is no effect on combat points.

(2) Deserts. Desert terrain degrades all forms of communications by 20%. Moving across desert terrain costs a unit one movement point per hex. During combat there is a one point advantage for the attacker in terms of combat points.

(3) Mountains. Mountains have the most serious effect on communications of all terrain types. They limit the tactical communications range to one hex, degrade ground wave HF by 40%, sky wave HF by 20%, but have no effect on satellite communications. If there are two or more hexes located less than three hexes from the unit's end of the path, no line-of-sight communications are possible. If the

mountains are three hexes from the end they will degrade the LOS link by 25%. It costs a unit three movement points per hex to move through mountains. A defender in the mountains receives a two-point advantage in combat.

(4) Ports and Mines. The city hexes degrade tactical communications by 20%, HF groundwave by 40%, HF skywave by 20%, LOS by 15%, but have no effect on satellite communications. A unit moving through a city hex will lose one mobility point per hex. A unit defending from a city will have a one point advantage in terms of combat points.

(5) Woods. Woods degrade tactical communications by decreasing the range by one half, degrade ground wave HF by 20%, and have no effect on LOS or satellite communications. Units moving through the woods lose two mobility points per hex. A unit defending in the woods has a one point advantage over the attacker.

(6) Lakes. Lakes degrade tactical communications by 15%, degrade all HF communications by 20%, degrade LOS communications by 15%, and have no effect on satellite communications. Movement across lakes is not possible. It would cost a unit 999 mobility points per hex, a number that no unit has. Since no unit can be located in a lake hex there is no value for combat points.

(7) Seas. Seas degrade tactical communications by 15%, degrade all HF communications by 20%, degrade LOS communications by 15%, and have no effect on satellite

communications. The only units that can move on seas are ships and marine units embarked on ships. Movement per hex costs one mobility point. There is no combat effect in combat points.

(8) Highways. There is no hex type for highways but highways do have an effect on mobility. If a unit is moving on a highway it only costs the unit a half of a movement point per hex.

b. Operations Game Play

This is the portion of the game where units are actually maneuvered to try to attain goals and engage in combat with opposing units. The rules of engagement (ROE) are established by the NCA and can be changed at the beginning of any turn. The ROE can be set so (1) the NCA must approve all attacks, (2) the JTFHQ can approve attacks, or (3) the local commanders can approve attacks.

There are eight options available that are briefly described below that control the actions of the units.

(1) Review Position and Goal Data. The first option available to the player is the option to review unit positions. If this option is selected, the player is given a table that shows that side's units, the hex coordinates of the units, the goals of the units (where the units are headed), and the combat points for each of the units.

(2) Set / Change Goal for a Unit. The second option enables the player to set or change the goal for a particular unit. When a new goal is selected for a unit a subroutine calculates the optimum route to the new location and displays the information on the screen.

(3) Change Electronic Equipment Missions. The next option available to the player is to change electronic equipment missions for systems.

(4) Assign Recon Missions. The next option allows the player to assign reconnaissance missions. The player can request photo recon missions for aircraft or can select satellites if satellite systems were available in the allocation phase.

(5) Assign Other Air Missions. The next option allows assignment of other air missions. The types of missions available are interdiction (up to five targets can be designated), counter air, and close air support.

(6) Embark/Disembark Marine Forces. The next option allows the marine forces to be embarked or disembarked. To be able to utilize this option, the marine force to be embarked must be on a hex adjacent to the amphibious task force hex. If the marines are to be disembarked, the amphibious task force must be adjacent to a land hex.

(7) Airlift Airborne Forces. The next option gives the player the ability to airlift the airborne forces.

To do that, the airborne forces must be located at an airbase. If they are airlifted, they can move to almost any hex on the board in one turn.

(8) Employ Anti-satellite Weapons. The final option for the player is to employ antisatellite weapons if available. If the player decides to use antisatellite weapons, a request must be made to the NCA. If approved, the weapon is employed at the beginning of the next turn.

When both players are finished with their turn or when the time runs out for the turn, control passes over to the umpire. At this point the umpire reviews the positions of the opposing forces, approves or disapproves the use of antisatellite weapons, changes the rules of engagement, and sets the length of the next game turn.

The umpire can also stop the game at this time if desired even though the results of the game are inconclusive. If the umpire decides to continue the game, the length of the game turn is entered and the next turn begins.

III. COMEL CORRECTIONS

Several COMEL software errors were encountered that had to be rectified before the graphics portion of the COMEL game could be instituted.

The manual gameboard did not match the default data files completely. Two of the most important terrain features on the gameboard are the two mountain passes. One of the mountain passes was depicted on the gameboard in the wrong location, causing the players to position units incorrectly and confusing the players on the capabilities of their units. The data file was corrected so that it matched the manual gameboard.

Some of the other terrain features were also depicted incorrectly. Different types of terrain have different effects on the units with regard to their speed and communications abilities. The gameboard showed open terrain in some areas while the data base indicated desert terrain. In some instances the hexes denoting the sea were improperly located, preventing the players from boarding their Marine amphibious forces on the ships.

The locations of the rivers in the data file were also improperly positioned. The locations of the rivers probably would not cause any major problems with the conduct of the game, but for accuracy the data file should reflect the

board that the players are using. After noting the discrepancies, the data file was altered to match the manual gameboard.

Another software problem was encountered in the COMEL command file. One of the options in that file enables the umpire to run the game using a tailored map file. When a different map file was created and the tailoring routine was executed, an error message was displayed. The error was located and corrected, allowing alternate map files to be inserted in the game.

IV. GENERAL APPROACH TO GRAPHICS FOR COMEL

There were several requirements that had to be achieved to convert the computerized version of COMEL using the manual gameboard to the fully automated version using the graphics display.

The most important capability for the graphics version was the ability to read the COMEL data files and properly display the information. The graphics enhancement had to be flexible enough to be able to read any valid data base that a player may design. The gameboard size must be constant, but the terrain attributes, the rivers, and the roads can all be different depending on the desires of the designer.

The same flexibility had to be made for displaying units on the gameboard. The default game only has ten units for each team. When the game is played, communications equipment can be dispersed. If it is, detachments are created which must be displayed on the board. The current version of the graphics enhancement will show up to twenty different units for each team, allowing the creation of as many as ten detachments. It also allows designers to create new units in addition to the default game units.

A. THE DI-3000 GRAPHICS PACKAGE

The graphics enhancement for COMEL could have been accomplished without using the DI-3000 graphics package by downloading operations codes directly to the Ramtek hardware. The DI-3000 tools were used because they are device-independent; i.e., the enhancement will require very slight modification if a graphics output system other than Ramtek is used. Even more, it contains Fortran-callable, high-level routines; i.e., many operations that normally require several commands are accomplished with one call to a DI-3000 routine.

There are two specific drawbacks to using the DI-3000 system. One is that when one color is written on top of another color, a third color results. This creates difficulties when trying to display one color object over another.

Another is that some of the DI-3000 routines that are listed in the user's manual do not execute properly. If a call is made to such a command an error message is generated when the program is linked to the DI-3000 library.

B. BASIC GAMEBOARD

The basic gameboard is created by accessing three different data files. They are HEX.DAT, VHEX.DAT, AND RHEX.DAT, each of which is a matrix data structure consisting of sixty-six rows with thirty characters per row.

1. HEX.DAT

This data file contains the basic terrain attributes for the gameboard. There are seven types of hexes used in COMEL.

a. Open Terrain

The open terrain is depicted in the data file by using the letter 'o'. On the manual gameboard, open terrain is shown as yellow. The graphics display shows open terrain as light green.

b. Deserts

The desert terrain is depicted in the data file by 'd'. The gameboard shows desert terrain as pink. The graphics display shows desert terrain as yellow.

c. Mountains

Mountains in the data file are represented by 'm'. On the gameboard, mountains are shown as brown hexes. The graphics display also shows the mountains as brown hexes.

d. Ports and Mines

The ports and mines in the data file are shown as 'c'. On the map they are shown in red, and are red on the graphics display. The data file was modified to indicate the locations of the cities, airbases, and forts by using 'c' for the graphics display. On the gameboard the cities, airbases, and forts are shown by outlining the

hexes, but on the graphics display they appear better when they are shown in red.

e. Woods

Woods are shown in the data file by 'w'. On the map they are shown in green and on the graphics display they are shown in dark green.

f. Lakes

Lakes are depicted in the data file by 'l'. On the map they are shown in light blue. The graphics display also shows them as blue.

g. Seas

Seas are shown in the data file by 's'. The map shows them as blue, as does the graphics display.

2. VHEX.DAT

This data file is used to show the locations of the rivers. The file consists of '0', '1', and '2' to show where the rivers run. Rivers can only run on the boundaries between hexes, so the graphics program had to read the data file and draw the rivers between the appropriate hexes.

3. RHEX.DAT

This data file is used to show the locations of the roads. The file consists of '0' and '1'. Roads only run from the center of one hex to the center of one of the six adjacent hexes. The graphics program only had to connect the centers of the hexes that had '1's in them.

C. UNIT DISPLAYS

The units are displayed on the basic graphics gameboard. They are accessed from within a subroutine that writes unit names and locations to the alphanumeric display. Once a unit and its location are defined, a subroutine is called that associates the unit with an alphabetic character. A white solid circle containing the unit's alphabetical character is then placed in the appropriate hex on the graphics display.

D. APPROACH TO GRAPHICS

Throughout this section, DI-3000 subroutine calls are used. Rather than explain each one as it called, all of the routines are explained in the maintenance manual (appendix 2). The general format for a DI-3000 command is CALL JXXXXX (parameters) or CALL KXXXXX (parameters). Recall that DI-3000 provides a library of FORTRAN-callable graphics subroutines.

The initial approach used to display the gameboard was to draw the hexes required for the gameboard one hex at a time. The center for the initial hex was calculated with respect to the world coordinate system and subsequent hex centers were calculated relative to that center. If virtual coordinates were used, the zoom and panning options that will be discussed later in the suggested enhancements section would not be possible.

Once a hex center was located, the coordinates for the six points on the sides were calculated. The next step was to connect the points for the hexes by using JMOVE to move to the coordinates of the first point and then call JDRAW to connect the first point with the other five points.

It was noticed that the coordinates of the centers of the hexes do not have to be calculated each time the program is run since they remain constant. To reduce run time the centers were precalculated and written to a data file called CENTER.DAT so they could be accessed whenever the plot routine was run. The calls to JMOVE and JDRAW were replaced by using JRPLGN which is used to full the interior of the hex with the appropriate color.

The interior color of each hex is determined by reading a data file called HEX.DAT. HEX.DAT specifies the different types of terrain used on the gameboard by using a single character in a matrix. A routine reads each of the characters from HEX.DAT and translates the character value into a numerical value that the DI-3000 subroutine JCOLOR uses as an index in a color lookup table. The color index is then written to a matrix called HEXCOLOR and is used in a call to JPIDEX.

When the interior color of the hex is specified another value is also determined and is written to a seperate matrix called BORDER. That value defines the color of the border to be used when the hex is drawn (edge attribute). When

JRPLGN is called it draws the hex using the specified border color. The interior style of the hex (color-filled) is specified by the JPIDEX subroutine.

Once the basic gameboard is displayed, the rivers are drawn. Drawing the rivers was somewhat more complex because each hex has to be examined to determine the locations of the rivers with respect to the surrounding six hexes. Rivers only run on the common sides between adjacent hexes, not through the hexes.

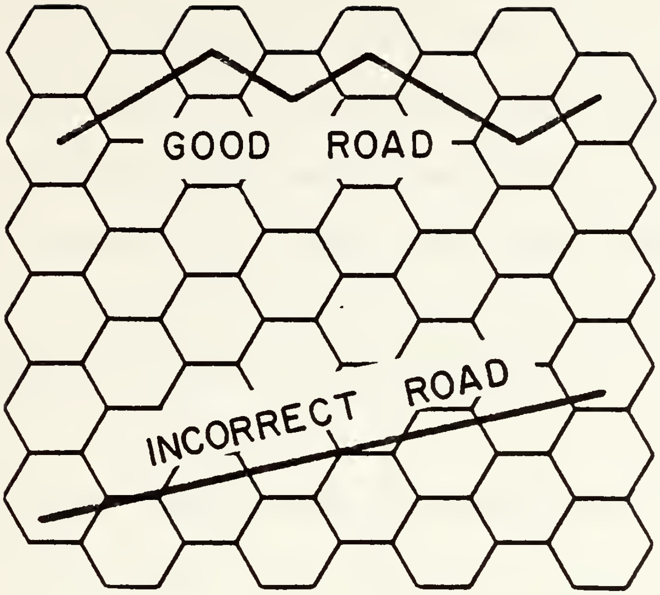
To draw the rivers a data file is accessed showing the locations of the rivers. If a hex has a value of 0 assigned to it none of the six sides of the hex contains a river. If a hex has a 1 or a 2 assigned to it, a river does run along one or more of the sides. If a hex has a value of 1 and any of the surrounding hexes also has a value of 1, there is no river running between them. The same holds true for adjacent hexes with each hex having a value of 2. If adjacent hexes have different values (1 and 2) a river runs on their common boundary. A data example and its associated output are shown in figure 2.

The initial approach to drawing the rivers was to examine each of the six sides of the hex to determine if that hex had a river on its boundary, but it was noticed that such an approach would cause duplication of effort on all interior hexes and some duplication of effort on the hexes on the edges of the gameboard. The modified check

routine only used three edges of the hex, decreasing the search time by half. After each of the three lower sides of a hex is compared to the adjoining hex, a graphics routine is called that draws the river on the common boundary if a river exists.

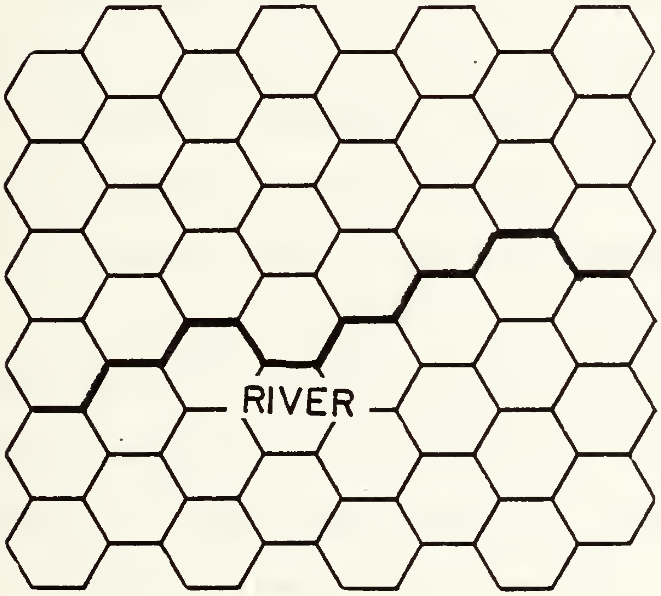
After the rivers are drawn, the roads are drawn. The same general approach is used that was used in drawing the rivers. The road data array differs slightly from the river data array. If a hex has a value of 0 there is no road running through it. If a hex has a value of 1 it does have a road running from its center to the center of an adjacent hex. The road can only pass from the center of one hex to the center of an adjacent hex. The road cannot go from one hex to another one without passing through the center of one of the adjacent six hexes (figure 3). The three lower sides of each hex were examined to determine if a road went from the hex being examined to one of the three adjacent lower hexes. If there is a road, a subroutine is called that draws the road.

Drawing all of the objects of the basic gameboard using the above procedure requires over two minutes. The basic gameboard can be saved as pixel data (the Ramtek displays 1024 x 1280 pixels on its monitor) via section file storage and retrieval routines that were created by Lawrence Livermore Laboratories.[Ref 2] The section "file" routine enables a graphics image in the Ramtek processor to be saved



```
 0 0 0 0
0 1 1 0
 1 1 1 1
1 0 0 1
 0 0 0 0
0 0 0 0
 0 0 0 0
0 0 0 0
 0 0 0 1
0 0 1 1
 1 1 1 0
1 1 0 0
 1 0 0 0
```

Figure 2. Tracing Roads Through Hexes.



```
 0 0 0 0
0 0 0 0
 0 0 0 0
0 0 0 2
 0 0 2 2
0 2 2 1
 2 2 1 1
2 1 1 0
 1 1 0 0
1 0 0 0
 0 0 0 0
0 0 0 0
```

Figure 3. Tracing Rivers Through Hexes.

pixel by pixel to a file that can be recalled later by using the "retrieve" routine. Since the only graphics output that is common to all three displays (red player terminal, green player terminal, and umpire terminal) is the display showing the basic board, the roads, and the rivers, the section file routine is called after those three elements are drawn and the composite image from the Ramtek is written to a map file.

A call to the section retrieval will recall the basic map with the roads and rivers on it and display the image on the Ramtek display within approximately fifteen seconds.

The first three graphics outputs (the basic board, the rivers, and the roads) were relatively easy to display because that information was displayed to each of the players' display terminals (Red, Green and Umpire). The remaining information is displayed to specific terminals and not to all three. Some output is displayed only to the red player or to the green player while other information is displayed to the red player or to the green player with the composite information displayed to the umpire.

A series of flags was created to select the appropriate Ramtek displays depending on the information to be displayed. By using the flags output could be sent to any individual display monitor or to any combination of them.

When the operations phase of the game is entered, each of the players' units that currently exists is called in

turn with its respective location. As the unit's location is read from a data base and the information is displayed on the player's terminal, a subroutine is called that plots the location of the unit on the graphics terminal.

The simplest way of depicting the units on the screen was to use an alphabetic character. It was found that just locating a letter on the hex made that letter hard to read, even with the line width of the characters set to maximum width. One solution to the problem would be to change the colors of the letters to provide more contrast to the hexes, but this would have meant accessing a data file again and would have increased the processing time. The solution used was to draw a white circle in the hex first and then draw the letter in the circle. The circle is small enough so the attribute of the hex can be seen but it is large enough so the unit letter can easily be seen (figure 4).

Another problem that was encountered was deciding how to display multiple units in the same hex. When one unit was located in a hex with another unit, its unit designator was overlayed on the other (see figure 5). To overcome that a routine was written that checks the location of each unit as it is accessed from the data file.

If a unit is being located in a hex where there is no other unit, it is plotted in that hex. If the unit being accessed from the data base is colocated with another unit, the new unit is offset one hex. In addition to offsetting

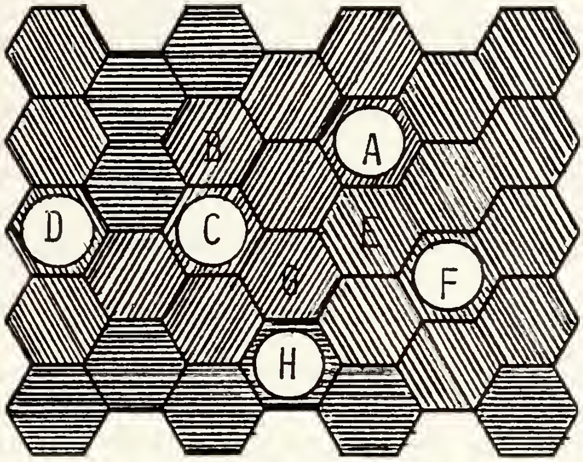


Figure 4. Increasing Contrast with Circles.

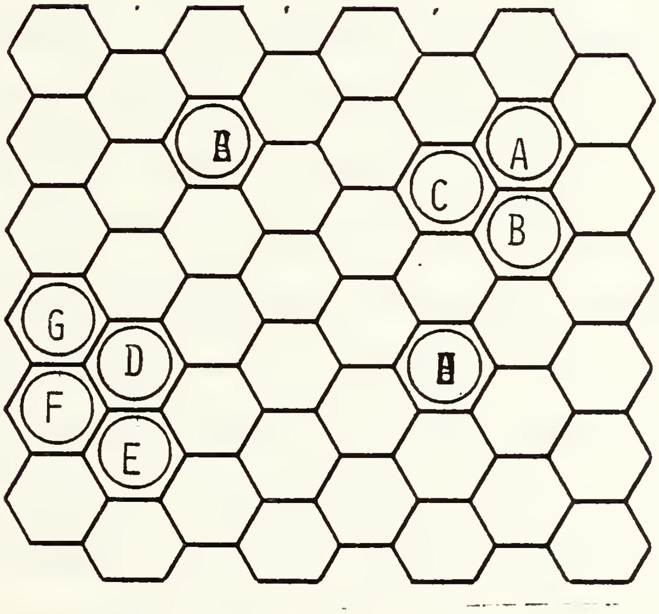


Figure 5. Offsetting Unit Designators.

the unit, the hex that it is plotted in is set to blink to indicate that the unit is not actually located in that hex. Even though the unit is shown in a different hex than the one it is actually in, the data file is not affected. It is possible that a blinking hex could be located adjacent to two non-blinking hexes but the frequency of such an occurrence is so remote that no special routine has been set up to clarify where the blinking unit is actually located. If an ambiguity did exist it could be solved by viewing the unit location menu on the alphanumeric display.

The Rowe-Allgood COMEL game has routines that will tell the player when a unit is in possible contact with enemy forces but does not tell the player in which hex the enemy unit is located. The graphics enhancement will highlight the enemy hex by making it flash but will not display the type of enemy unit that is actually there.

The same is true for surveillance reports. Previously the report from aircraft and satellites informed the player where an enemy unit was located. The player then was required to manually mark the enemy unit's location on the map. With the graphics enhancement, the hex that an enemy force is located in will flash. Each time a new hex is designated as an enemy hex, that hex will also flash. By observing which hexes are flashing and in which order they are set to flash a player can track an enemy unit.

The enhancements implemented above allow play of COMEL without the manual gameboard. All facets of COMEL game play are now accomplished through keyboard, alphanumeric and graphics interaction with the computer.

V. RECOMMENDATIONS

This thesis dealt only with the graphics enhancement of the operations phase of COMEL and did not add any graphics capabilities to the other two phases (acquisition and allocation). Possible future enhancements for COMEL are discussed in this section.

A. ENHANCEMENTS FOR THE ACQUISITION PHASE

The acquisition phase of the game uses several bar charts to display information but several menus must be accessed to review the charts. One graphics enhancement that would reduce look-up time would be to display all of the charts on the Ramtek. With the high resolution of the display each chart could be displayed simultaneously. The effects of decisions could be displayed immediately. Analyzing the effects of such decisions is possible now but the process is slow and tedious. The graphics presented currently are very crude.

B. ENHANCEMENTS FOR THE ALLOCATION PHASE

During the allocation phase of the game communications equipment is allocated to units for use during the operations phase of the game. Different systems must be compatible for each to be effective. A table of communications equipment shows which systems are compatible

with other systems. Reading this table is tedious. The tables may be misread and systems may be misallocated. During the conduct of the game there would be no way of knowing that the lack of communications was due to misreading the allocation chart.

Graphics enhancement could help by displaying network lines showing the effectiveness of a particular link. A piece of equipment could be allocated to one unit and another piece of equipment to a second unit. The compatibility of that link could be checked by using the graphics capability. The capability to allow changing the allocation of equipment during this phase would allow investigation of alternate allocation schemes.

C. FURTHER ENHANCEMENTS FOR THE OPERATIONS PHASE

The graphics enhancement of the operations phase at this time shows the location of the units for the red and green forces and highlights the hexes where conflicts occur. There are other enhancements that can be made for the operations phase of the game.

1. Symbols

One of the additional enhancements that can be made to the operations phase of the game is to use iconic representations to represent the units instead of the alphabetic characters currently being used. Such an

enhancement would entail creating and saving the icons in a file and recalling them when appropriate.

There are several methods that could be used to depict the units. One would be to create standard military map symbols to display a particular unit. However, the small hex size might not allow adequate display of the symbols.

Another method is to let an iconic symbol represent the unit. An airplane could represent the different tactical fighter wings and small tanks could represent the armor units. However, a method of distingushing between like units would have to be established. To display a graphic symbol and an identifier in the same small hex might make it difficult to read.

2. Zoom and Pan

A graphics enhancement that could resolve the above display problems would be the capability to zoom in on a particular window of the map, amplifying the size of the hexes by as much as eight times. The zoom subroutine is an option available on the DI-3000 graphic package. The zoom subroutine effectively either changes the JWINDO parameters so the real world coordinate limits on the display screen are changed or it replicates pixels. The zoom function does not affect the data or the section files.

In conjunction with the zoom capability is the ability to pan around the display area. The panning ability

is made possible by the use of pick inputs in the form of a graphics tablet and a cross-hair cursor device. The upper left corner of the tablet represents the upper left corner of the virtual coordinate display area. The other three corners of the graphics tablet represent the other three corners of the virtual coordinate map.

By using the zoom and pan options concurrently the player can expand the map display and the units on it so that symbols on the display can easily be read.

3. Highlightability

A further enhancement for the operations phase of the game would be to highlight satellite tracks so the player knows exactly what portion of the map the surveillance satellite is covering. The coverage could be shown by using flashing hexes or by changing the color of the hexes.

One way to show the coverage would be to open a retained segment for the satellite track. A retained segment is given a segment number when it is drawn. It can be overlayed over a retained segment with a lower segment number. Such a segment could be displayed each time the map board was displayed. If the satellite track is changed that particular retained segment could be deleted and recreated using the new coverage track.

A different way to display the track would be to use a timer that would allow the display to be visible for a

certain time period and then delete the segment. A major advantage of deleting the segment would be to keep the screen from getting cluttered up with information that is not required constantly. The option could be added so the player could have the satellite tracks displayed at any time.

4. Graphics Tablet

Utilization of the graphics tablet and the cursor input devices to move the unit would speed up game play. The way the game is currently played the player must manually enter at the keyboard the destination hex coordinates of each unit to be moved. There is a check routine in the game that prevents the player from using the wrong hex coordinate pairs (the coordinates must both be odd or even) but that does not prevent the player from accidently placing units on top of a mountain or in the middle of a forest. Using the input devices to select the destination of units to be moved will reduce error and be more user friendly. It will also speed up the play of the game.

5. Lines of Bearing

When EW units are operating in the ESM mode, lines of bearing to enemy units within range of the EW units are displayed on the alphanumeric terminal. It would be more

beneficial to be able to see the lines of bearing displayed on the graphic display monitor.

6. Communications Connectivity

Communications graphics enhancement would be the capability to display the communications connectivity for each unit. Such a display could show the probability of communications for each of the communications models for each unit displayed as a bar chart so the relative capabilities for each unit could be assessed. If one unit had a high capability in one area and another unit was low in the same area, some equipment from the one unit could possibly be reallocated to the other.

7. Optimum Tracks Graphically Displayed

The current version of COMEL displays the optimum track on the alphanumeric display screen, giving the hex coordinates. the player must manually plot the hex locations to see the track. A graphic display of the track would enable the player to see the proposed track and quickly comprehend the unit movement.

As more experience is gained by playing COMEL more input can be generated for possible further enhancements.

LIST OF REFERENCES

1. Rowe, Katherine and Allgood, Robert C. Jr., COMEL: A Communications War Game , M. S. Thesis, Naval Postgraduate School, Monterey, California, 1983.

2. DI-3000 User's Guide, Precision Visuals, Inc., Boulder, Colorado, 1982

3. Digital Software, VAX/VMS Guide to Using Command Procedures, (Order No. AA-H782B-TE) Digital Equipment Corporation, Maynard, Massachusetts, 1982

4. Digital Software, VAX/VMS Command Language User's Guide, (Order No. AA-D023C-TE) Digital Equipment Corporation, Maynard, Massachusetts, 1982

5. Digital Software, VAX-11 FORTRAN Language Reference manual, (Order No. AA-D034C-TE) Digital Equipment Corporation, Maynard, Massachusetts, 1982

6. Digital Software, VAX-11 FORTRAN User's Guide, (Order No. AA-D035C-TE) Digital Equipment Corporation, Maynard, Massachusetts, 1982

7. Digital Software, VAX/VMS Primer, (Order No. AA-D030B-TE) Digital Equipment Corporation, Maynard, Massachusetts, 1980

APPENDIX A - SECOND EDITION OF THE USER'S MANUAL

The purpose of this manual is to supplement the user's manual that was written for Communications Electronics War (COMEL) [Ref. 1] with documentation of the graphics enhancement described in the thesis. The original user's manual provides an introduction to the game, communications and electronics effects, umpire game play, and the player's game. It provides very detailed instructions in each section so the red and green players and umpire can thoroughly understand the game.

This supplemental manual will briefly describe the topics that were covered in the original manual and will discuss in detail the changes that were made as a result of the graphics enhancements described in the thesis.

A. MAJOR DIFFERENCES

This section will discuss the differences that the players will encounter when playing COMEL with the graphics enhancement.

1. The Gameboard

The original manual COMEL gameboard is a board showing a hexagonal matrix of sixty-six rows and thirty

columns. The hex colors indicate the hex attributes. There are six colors on the manual gameboard.

The graphics gameboard is very similar to the manual gameboard in that it is also a hexagonal matrix of sixty-six rows and thirty columns. The graphics gameboard, however, has a seventh color (a second blue) to differentiate between the lakes and the sea. Additionally, the cities, forts, and airbases are shown in red to make them easier to see.

2. The Game Pieces

The manual gameboard uses small cubes of wood to represent the units. Each cube has a small magnet glued to the bottom to keep the piece in place when the game is being played. The pieces being used to designate units have small flags with numbers on them to designate different units. There are also pieces for the opposing forces without flags to use when a hostile unit is detected. If more than one unit is located in a hex the player must decide how to depict that on the board.

The graphics version of the game uses letters of the alphabet to designate units on the grahics display screen. The letters are written in a white circle to make them easier to see against the hexes. If more than one unit is located in a hex the first unit is shown in that hex. The other units are shown in adjacent hexes with the white circles flashing to indicate that the units are not actually located in those hexes.

3. Unit Movement

Unit movements for both versions of the game begin the same way. When a turn begins the location of each unit is displayed on the terminal. The board must then be surveyed to determine exactly where each game piece is located. If a unit has moved, the new hex location must be located and the game piece must be moved to that location. In the manual game this is a time-consuming, error-prone operation.

Using COMEL graphics enhancement the units are shown on the graphics display using the information that is still displayed on the terminal. That means as soon as the information is displayed on the terminal it is also shown on the graphics display. The board no longer needs to be surveyed to determine which pieces must be moved. The movement is done automatically so there is no chance for errors.

4. Enemy Locations

When a surveillance satellite or reconnaissance aircraft locates an enemy unit or when friendly and enemy units move into adjacent hexes, the hex that the enemy is located in is highlighted by a flashing hex. The identification of that unit is not displayed. The flashing hex will not be deleted even if a new one is identified, so it will be possible to track the movement of an enemy unit.

B. RED AND GREEN PLAYERS

The following sections provide a summary of the procedures and directions to play COMEL. See Reference 1 for a more detailed discussion. When ready to run COMEL, find a terminal that is connected to the VAX/VMS system in the WAR Lab.

When successfully logged in to COMEL and the "$" prompt appears, type '@PLAYER'. The following display will appear:

```
----------------------------------------------------------
:                                                        :
:   CCCCC                                                :
:  CCCCC      0000                                       :
: CC         000000    M     M                           :
: CC        00    00   MM   MM   EEEEEEE                 :
: CC        00    00   MMM MMM   EEEEEE    L             :
:  CCCCC    00    00   MM M MM   EE        LL            :
:   CCCCC    000000    MM   MM   EEEEEEE   LL            :
:             0000     MM   MM   EE        LL            :
:                      M     M   EEEEEE    LL            :
:                                EEEEEEE   LLLLLL        :
:                                          LLLLLLL       :
:                                                        :
----------------------------------------------------------
YOU CAN ENTER THE CARRIAGE RETURN AT ANY TIME.
UMPIRE PERMISSION IS NOT REQUIRED.
(C/R)
```

When the return key is pressed, a menu of options will be displayed.

```
*-------------------------*
*  COMEL PLAYER OPTIONS   *
*  CHOOSE:   ##           *
*-------------------------*
   ##  ACTION REQUESTED
   --  ----------------------
    1  GAME INFORMATION
    2  RED    PLAYER?????
    3  GREEN PLAYER?????
    4  ACQUISITION PHASE ONLY
    5  OPERATIONS  PHASE ONLY
    6  BOTH PHASES
    E  EXIT GAME
##
```

1. Game Information

The first option will present game information. The prompt will ask for a number between 0 and 24, a number which will give a certain page of information. If a 0 is entered, a prompt to press the return will appear and the program will return to the menu.

2. Red Player

The red player selects this option. If both players select this option the game will not run properly.

3. Green Player

The green player selects this option. As with the red option, both players cannot select the green option.

4. Acquisition Phase Only

This option would be selected if the acquisition phase only is being run, and after selecting either the red or green player. Coordination must be made with the other player and with the umpire to make sure all game players are

using the same option. When this option is selected the following banner will be displayed:

```
*********************************************************
*  CCCC  OOO  M   M EEEEE L                             *
* C     O   O MM MM E     L                             *
* C     O   O M M M EEEE  L                             *
*  CCCC  OOO  M   M EEEEE LLLLL                         *
*                                                       *
*   A    CCCC  QQQ  U   U I SSSSS I TTTTT I  OOO  N   N *
*  A A  C     Q   Q U   U I S     I   T   I O   O NN  N *
* AAAAA C     Q   Q U   U I  SSS  I   T   I O   O N N N *
* A   A  CCCC  QQQ   UUU  I SSSSS I   T   I  OOO  N   N *
*                                                       *
* SSSSS TTTTT   A    GGGG EEEEE                         *
* S       T    A A  G     E                             *
*  SSS    T   AAAAA G GGG EEEE                          *
* SSSSS   T   A   A  GGGG EEEEE                         *
*********************************************************
DO NOT ENTER THE CARRIAGE RETURN UNTIL
INSTRUCTED TO DO SO BY THE UMPIRE
(C/R)
```

Once the acquisition stage is actually entered, the following menu will appear:

```
  COMEL ACQUISITION STAGE
  ----------------------------
 (YEAR  1 : BUDGET $  500.0M)
 (       OBLIGATED $    0.0M)

 *DECISION TIME:  27 MIN*
:------------------------:
:CHOOSE: FROM ##         :
:------------------------:
  ##  TITLE
  --  ---------------------
  1   SHOPPING LIST
  2   SYSTEMS STATUS
  3   INTELLIGENCE
  4   BUDGET ANALYSIS/REQ
  5   PROCUREMENT
  E   EXIT ACQUISITION
?
```

a. Shopping List

This option shows a shopping list of equipment and systems available for purchase. If the shopping list option is chosen, other options are displayed to get general information, technical information, and/or cost information on a specific piece of equipment or system.

b. Systems Status

When this option is chosen, a menu is displayed that makes available an updated systems summary, a list of systems deployed, systems in R&D, systems that have finished R&D and are ready for M&D, systems in M&D, and systems available to be bought.

c. Intelligence Reports

This option provides intelligence reports on the status of the opposing force's systems status. To get the intelligence a request for the intelligence must be sent to the umpire. The request costs money. The umpire has the option of giving the player no information, a little bit of information, or a great deal of information.

d. Budget Analysis/Request

The budget analysis/request displays a five year cost projection and allows the budget request for the next year to be submitted. By using this option information can be obtained to allocate resources to have equipment developed normally or have it placed in advanced R&D so it can be available sooner. Requests for a budget for the next

year are also submitted. The umpire has the option of allocating the amount that has been requested or allocating more or less than was requested. The program is designed so the umpire cannot give the players too large or too small a budget.

e. Procurement

This option deals with the procurement of the equipment and systems. When selected help can be obtained thru a decision analysis aid, systems can be bought, systems can be scrapped (if the remaining funds are running low), equipment and systems in R&D can be placed in accelerated R&D, the R&D effort can be expanded, O&M can be reduced, intelligence can be purchased from the umpire or messages can be sent to the umpire, and systems can be placed in M&D.

f. Exit Acquisition

When selected, the main game menu will return.

5. <u>Operations Phase Only</u>

This main menu option is selected to run the operations phase only. Again, coordination with the other player and the umpire must be made so all game players select the same game phase.

When this option is selected, the following banner will appear:

```
**********************************************************
*     ****     ****     *     *     ******    *          *
*    *    *   *    *    **   **     *         *          *
*    *        *    *    * * * *     *****     *          *
*    *    *   *    *    *  *  *     *         *          *
*     ****     ****     *     *     ******    ******     *
*                                                        *
*              ****     *****        ****                *
*             *    *    *    *      *                    *
*             *    *    *****        ****                *
*             *    *    *           *    *               *
*              ****     *            ****                *
*                                                        *
**********************************************************
Do not proceed past this point until told to do so by the
Umpire.  When the Umpire tells you to go on, enter (C/R)
```

When the operations phase is entered for the first time (not for a game that is being resumed), the equipment that was developed in the acquisition phase or the equipment that is provided from the default files must be allocated to units. When a unit name is specified, lower case letters only are used for the name; e.g., 35th abn.

Some large equipment will be assigned a fixed-station role and will be located in a particular hex.

If the equipment is mobile it can be attached to a combat unit and moved with it or it can be detached and set up as a seperate communications relay or EW unit.

Airborne command posts fly on a rotating schedule. If one was acquired, it will be available about 45% of the time. Two ABNCP's will be available about 75% of the time, and three or more will be available about 100% of the time. AWACS are scheduled in a like manner.

Communications satellites have either one area beam or several spot beams. The centers of the beams are directed at a particular hex with an effectiveness radius given.

Intelligence satellites can survey a number of adjacent columns. If any enemy units are located within those columns, a message will be printed out on the CRT giving the hex location but not the identification of the unit.

Anti-satellite weapons can be used against either communications or intelligence satellites. They are used once and only with the approval of the NCA (Umpire).

Once the equipment has been allocated, the following menu is displayed:

```
                Action Menu

     1.  Review Position and Goal Data
     2.  Set/Change Goal for a Unit
     3.  Change Electronic Equipment Missions
     4.  Assign Recon Missions
     5.  Assign Other Air Missions
     6.  Embark/Disembark Marine Forces
     7.  Airlift Airborne Forces
     8.  Employ Anti-satellite Weapons
     9.  Finished Turn

     Enter the number of your chosen action.
```

These options can be used more than once per game turn except for option 9 (finished turn) or until the time for the turn runs out. The options are briefly described below.

a. Review Position and Goal Data

Shows the current location, destination, and combat points for each unit.

b. Set/Change Goal for a Unit

Permits movement of combat units. When the goal for a unit is to be changed, the new coordinates are entered. Depending on the range, terrain, communications gear, and other factors connectivity to a unit will not be made and the requested order cannot be accepted. An example of a successful goal change is:

```
Which unit goal do you want to set or change?
Enter the unit's name - up to 8 spaces.  234 mech
The goal for the 234 mech is now 0 0.  Is this ok as is?
y or n
n
What is the desired goal?
Use 0 0 to stop the unit in its present position.
23 37
The goal of the unit has been changed to 23 37.
There will be a short delay while the route is
being planned.

The projected path is:
19 29   18 30   17 31   16 32   15 33   17 33   18 34
19 35   21 35   22 36   23 37
this path will take about 10 movement points.  If you
want to take a different route, resubmit a closer,
interim goal for this unit.
```

c. Change Electronic Equipment Missions

This option presents the following menu:

```
         COMM MENU
     1. Assign ABNCP and AWACS orbits
     2. Detach equipment for use as a relay or EW unit
     3. Change comm satellite orbit
     4. Change EW system mission
     5. Return to main menu
Input number of desired action.
```

The first option returns the availability of the ABNCP and AWACS. If one is available the center of the orbit is entered.

The second option detaches equipment and sets it up as a seperate detachment. The name for the detachment is then entered.

The third option returns the center focus for each communications satellite. The beam location can then be changed or left in its current location.

The fourth option designates the mission of the EW equipment to be either ECM or ESM.

The final option returns to the main menu.

d. Assign Recon Missions

This option permits satellite surveys or air reconnaissance missions to be assigned.

e. Request Other Air Missions

When this option is selected, several air missions can be assigned. The choice of air missions are interdiction, close-air-support, counter air, and no new missions. Each choice will give some information about the option.

f. Embark/Disembark Marine Forces

To be able to embark a marine brigade onto an amphibious task force, the marines must be in a hex adjacent a sea hex and the amphibious task force must be in that sea

hex. To disembark the marine brigade, the amphibious task force must be in a sea hex adjacent to a shore hex.

g. Airlift Airborne Forces

For an airborne force to be airlifted it must already be located at an airfield. Once the unit is loaded and dropped it must return to an airfield before it can be dropped again.

h. Finished Turn

When selected, the control module is notified and returns a message stating that the next turn will begin in about ten minutes. The umpire specifies the length of the turn but if both teams get finished before the time limit is up, the game can continue immediately.

6. Both Phases

If both phases of the game are to be played, select option # 6.

7. Exit Game

If this option is selected, the game will be terminated. The terminal will return to the system level and the "$" prompt will be displayed. If COMEL is to be played again '@PLAYER' must be retyped.

C. UMPIRE

The umpire uses the same procedure as the players to log in. Once successfully logged in the system prompt "$" will appear.

1. Main Menu

Once the prompt appears type "@COMEL". The following menu will appear:

```
COMEL WAR GAME OPTIONS
----------------------
: CHOOSE: ##         :
----------------------
##  ACTION REQUIRED
--  ----------------------------------------------------
 1  NEW GAME : DEFAULT  DATA/EQUIPMENT FILE EMPTY
 2  NEW GAME : DEFAULT  DATA/OPERATIONS  START
 3  NEW GAME : DEFAULT  DATA/ACQUISITION START
 4  OLD GAME : CONTINUE DATA/DESIGNATED SAVE AREA
 5  MOD GAME : TAILOR   DATA FILES
 6  SAVE   GAME
 7  DELETE GAME
 8  DELETE FILE
 9  REVIEW MAP FILE
 R  START  GAME
 E  EXIT   COMEL
```

All of the options listed are explained in detail in the original edition of the users' manual except for option 9. The original options will be summarized and option 9 will be explained in detail.

a. New Game : Default Data/Equipment File Empty

This option starts in the acquisition phase with no equipment ready for allocation. All equipment to be allocated to units when the operations phase begins must be acquired during the acquisition phase.

b. New Game : Default Data/Operations

This option provides a default set of equipment to be allocated to combat units when the operations phase begins. it provides typical command and control

capabilities to the players. The operations phase begins with this option.

c. New Game : Default Data/Acquisition Start

This option begins the acquisition phase with a rudimentary, default set of equipment already available to the units. During the acquisition phase more equipment can be acquired.

d. Old Game : Continue Data/Designated Save Area

Selecting this option will result in the following menu being displayed:

```
-----------------------
: OLD GAME SELECTION :
: CHOOSE:   ##        :
-----------------------
   ##  ENTER SAVED WAR GAME
   --  --------------------
    0  EXIT SAVED GAMES
    1  COMEL WAR GAME 1
    2  COMEL WAR GAME 2
    3  COMEL WAR GAME 3
    4  COMEL WAR GAME 4
    5  COMEL WAR GAME 5
    6  COMEL WAR GAME 6
    7  COMEL WAR GAME 7
    8  COMEL WAR GAME 8
    9  COMEL WAR GAME 9
```

This option assumes that a game is being resumed. When any number between 1 and 9 is selected the game files are loaded with the information from that respective game file. When the information has been retrieved and the game files have been updated, the main menu reappears.

e. Mod Game: Tailor Data Files

When this option is selected, the game can be tailored in many ways. Data files from different games can be retrieved or the map may be changed. The original thesis and the maintenance manual should be read before attempting to tailor data files.

f. Save Game

When this option is selected, the following menu will appear:

```
---------------------
: SAVE GAME TO FILE :
: CHOOSE:   ##      :
---------------------
   ##  ENTER GAME AREA FOR SAVE
   --  ------------------------
    0  EXIT  GAME SAVE ROUTINE
    1  COMEL WAR GAME 1
    2  COMEL WAR GAME 2
    3  COMEL WAR GAME 3
    4  COMEL WAR GAME 4
    5  COMEL WAR GAME 5
    6  COMEL WAR GAME 6
    7  COMEL WAR GAME 7
    8  COMEL WAR GAME 8
    9  COMEL WAR GAME 9
```

When options 1 thru 9 are selected the game information is from all files is saved into a master file. If a game was previously saved with the same number and a new game is saved with the same number, both games can still be retrieved. When the game files have been saved, the game returns to the main menu.

g. Delete Game

This option displays a listing of game files that have been stored. Any of the game files can be deleted.

h. Delete File

If this option is selected, the same options that appeared under Delete Game will be shown, but another option will appear that will allow a specific file from a version of a game to be deleted.

i. Review Map File

This is a new option that is used only with the graphics enhanced version of COMEL. When this option is selected, a listing of all of the map files will be displayed. These are the map files that can be entered later when the request to enter a map file is displayed. After the umpire assigns graphics display units to the players, the request will appear as follows:

```
 What monitor is Red using?
0
 What monitor is Green using?
0
 What monitor is the Umpire using?
3
 Do you want to retrieve a previously saved map file?
y
 Do you want to specify a map name?
 If not, the default COMEL map will be used.
n
```

The above answers to the prompts would result in the default COMEL game map being drawn on all initialized monitors.

If a new hex map is going to be created, just creating the data file is not sufficient. The map must be drawn on the Ramtek before it can be saved as a map file. A new map is created by making a copy of HEX.DAT and changing the hex attributes. If new rivers and roads are also desired VHEX.DAT and RHEX.DAT must also be copied and modified. As soon as the map has been drawn on the Ramtek, a prompt will ask for the map file name. When the map has been written to the map file, it will be available for retrieval.

j. Start Game

This option will start the game. If the game is terminated by entering Control Y or Control C, control of the game will pass back to the game menu. The information in the current turn is lost but any previous information is still in data files and can be saved by using the Save option.

When this option is selected, the following menu will appear:

```
*--------------------------------*
*      COMEL WAR GAME OPTIONS    *
*      CHOOSE:   ##              *
*--------------------------------*
     ##  ACTION REQUESTED
     --  --------------------------
      1  GAME INFORMATION
      2  RUN  ACQUISITION ONLY
      3  RUN  OPERATIONS PHASE ONLY
      4  RUN  COMPLETE GAME
      5  EXIT COMEL
##
```

The first option will provide game information. The next three options will begin the phase or phases selected. The last option will return control of the game to the main menu.

The umpire is an active participant in all phases of COMEL.

2. Acquisition Phase

When the acquisition phase is started, the following banner is displayed:

```
**********************************************************
*  CCCC  OOO  M   M EEEEE L                              *
* C     O   O MM MM E     L                              *
* C     O   O M M M EEEE  L                              *
*  CCCC  OOO  M   M EEEEE LLLLL                          *
*                                                        *
*   A    CCCC  QQQ  U   U I SSSSS I TTTTT I  OOO  N   N  *
*  A A  C     Q   Q U   U I S     I   T   I O   O NN  N  *
* AAAAA C     Q   Q U   U I  SSS  I   T   I O   O N N N  *
* A   A  CCCC  QQQ   UUU  I SSSSS I   T   I  OOO  N   N  *
*                                                        *
* SSSSS TTTTT   A    GGGG EEEEE                          *
* S       T    A A   G     E                             *
*  SSS    T   AAAAA G GGG EEEE                           *
* SSSSS   T   A   A  GGGG EEEEE                          *
**********************************************************

PLEASE ENTER A 4-7 DIGIT ODD NUMBER TO BE USED AS A SEED
FOR THE RANDOM NUMBER GENERATOR USED TO DETERMINE
THE START OF THE WAR:
```

The odd number is used to generate a random number to determine if the war has begun. War can occur at any time beginning in year two on.

The next request is for the length of the turn in minutes. The program will not accept a game turn of less

than ten minutes. If 0 is entered the game control will go back to the menu. The first game turn should be fairly long (suggested time is 90 minutes) so the players can become familiar with the different options. The second turn should be shorter (recommended 60 minutes) and subsequent turns should be even shorter (recommended 30 minutes). If both players finish their turns before the time has expired, the game will go on to the next turn.

There are several decisions that must now be made.

a. Budget

An annual budget is determined for each team prior to each year of the acquisition phase. To help the umpire decide how much money to allocate to each team, a five year budget summary is displayed upon request. One item to remember is that the intelligence requests cost each team between 10 and 20 Megabucks per request. Those requests are not shown in the budget summary so additional funds must be allocated. After the budgets are entered the program goes into the synchronization and control phase.

b. Synchronization and Control

The synchronization and control of the acquisition phase is controlled by the system. Once the return key is pressed the following message is displayed on the umpire's terminal:

```
WAITING FOR GREEN & RED...17:30:25
```

When the red and green players have been told to enter the acquisition phase of the game the display on the umpire's terminal will display the time remaining in that game turn. When the time has expired or when the players have finished their turns, the game will continue to the intelligence update routine.

c. Intelligence Update

As soon as the intelligence update routine is entered, the following menu will appear:

```
*----------------------------*
* INTELLIGENCE UPDATE MENU *
* CHOOSE ##                 *
*----------------------------*
    ##  ACTION REQUESTED
    --  ---------------------
     1  RED   GAME SUMMARY
     2  GREEN GAME SUMMARY
     3  READ  RED   MAIL
     4  READ  GREEN MAIL
     5  RED   INTEL REQUEST
     6  GREEN INTEL REQUEST
     7  UPDATE RED INTEL
     8  UPDATE GREEN INTEL
     E  EXIT INTEL
?
```

The first and second options will produce a menu that allows the status of the red and green players respectively to be reviewed.

The third and fourth choices will produce any mail from the red or green players.

The fifth and sixth options will display any intelligence request from the red and green players. If there is a request that means that the team that made the

request spent money from its budget. If the umpire decides not to provide any intelligence, the money was spent for nothing. A request can be made for specific or general intelligence with the level of intelligence returned being an umpire decision.

The next two options provide the capability to respond to the players' intelligence request. The response to the players is a free format narrative.

When the final option is selected and the return is pressed, either the game continues on to the next acquisition year or war begins. If a message is displayed on the umpire's terminal stating that war has begun, the umpire can override the beginning of the war. If war is allowed to begin, the operations phase begins.

3. Operations Phase Control

During the operations phase of the game the umpire controls the length of the game turns. For the first turn a decision must also be made to either begin a new operations phase or to resume a previously saved game.

If a new operations phase is being entered the umpire decides whether to use default equipment files or to build new equipment files.

At the beginning of each turn the umpire (NCA) decides on what level to set the rules of engagement (ROE). Initially, the NCA must approve any attacks. As the game progresses the NCA can change the ROE so the JTF commander

or the local commander can approve attacks. The NCA must always be consulted to approve the use of anti-satellite weapons.

At the end of each turn the NCA can stop the war. If one side controls the major and minor objectives that side is a decisive winner and the game ends. If neither side is a decisive winner, the NCA can declare a cease fire, save the files, and continue on later or the NCA can declare one of the teams a winner based on the current situation. Using the Ramtek graphic display the umpire can easily see the disposition of the units for both teams, making it easy to follow the progress of the teams.

D. SUMMARY

This manual is only a summary of the original COMEL user's manual and does not go into great detail on some of the options available to the red and green players and the umpire. This manual is being merged with the original user's manual which is available in the COMEL game file on the computer system located in the WAR Lab.

APPENDIX B - SECOND EDITION OF THE MAINTENANCE MANUAL

The purpose of this manual is to supplement the maintenance manual that was written for Communications Electronics War (COMEL) [Ref 1] with documentation of the graphics enhancement described in the thesis. The original maintenance manual documented the software used in COMEL so that it could be modified easily. The program was written using modular design, frequent comments, and structured FORTRAN in an attempt to write code that could be read, understood, and modified by novice programmers. Further information can be found in the COMEL User's Manual [Ref 1] and in the COMEL Thesis [Ref 1].

This supplemental manual will list each module that was modified to support the graphics enhancement package. Duplicate information not required to support the graphics has been omitted.

This manual follows the same structure as the original manual so a person familiar with the original manual can understand the supplement without learning a new format.

A. HARDWARE/SOFTWARE DEPENDENCY

This section describes the characteristics of the program that make COMEL and the graphics enhancement

dependent on the specific computer and operating system for which it was designed.

COMEL was developed on a Digital Equipment Corporation VAX 11-780. The operating system used is VMS. The graphics portion of COMEL was written using Precision Visual's DI-3000 graphics software, FORTRAN-77, and the RAMTEK 9460 graphics system. The DI-3000 graphics system was designed to be hardware independent so it can be used with other types of graphics display systems.

FORTRAN-77 programming constraints such as the IF-THEN-ELSE-IF, are used extensively, as is variable type character which allows alphabetic responses to questions with direct error checking. Such FORTRAN constructs permit easy structuring of the program and enhances its maintainability and readability. To modify the game program to compile with other dialects of FORTRAN that do not support these features would require a line-by-line review of the code to locate and modify these structures.

Nearly all of the graphics routines rely on the DI-3000 FORTRAN-callable routines. The DI-3000 package provides an extensive library of graphic subroutines that are easy to use. If COMEL is run on a system that does not have a DI-3000 library, the program would have to be examined line-by-line to determine where the DI-3000 calls are located and to determine what routines would have to be written to accomplish the same results.

B. GRAPHICS ENHANCEMENTS

1. COMMON VARIABLES (GRAPHICS ENHANCEMENT)

The graphics enhancement uses eight COMMON statements in addition to the ones used in the original version of COMEL. The common statements and their respective variables are identified below.

COMMON /CONFLICT/ conunt

conunt - integer; a variable that is used to increment the counter for segment numbers when accessing a routine that highlights hexes that have possible conflict situations. It indicates a CONflict UNiT.

COMMON /COPY/ mapname

mapname - character; identifies the map being used for the display. When the display is erased mapname allows the map to be redrawn without the umpire reentering the name.

COMMON /COPY1/ copyflag

copyflag - logical; signifies that SUBROUTINE RETRIEVE is being accessed from SUBROUTINE DELETE and not from SUBROUTINE HEXX. Copyflag passes the control of the program over the questions found in the first part of SUBROUTINE RETRIEVE.

COMMON /DELETE/ maxx,maxy

maxx - integer; a counter that keeps track of the number of red units that have been displayed. If the number of red units is not accurately tallied, the call to JPURGE

will result in an error message if the count is too high or units being left on the screen if the count is too low.

maxy - integer; a counter that keeps track of the number of green units that have been displayed. If the number of green units is not accurately tallied, the call to JPURGE will result in an error message if the count is too high or units being left on the screen if the count is too low.

COMMON /GRAPHICS/ ratio, iriver, iroad, x1, y1

ratio - real; a variable that is used by the DI-3000 to set the display window size (JWINDO). The value of ratio is determined by using a routine called JASPEK.

iriver (66,30) - integer; an array that designates which hexes have rivers on their boundaries.

iroad (66,30) - integer; an array that designates which hexes have roads running through them.

x1 (66,30) - real; the x-coordinate for the center of the hex

y1 (66,30) - real; the y-coordinate for the center of the hex

COMMON /MONITOR/ rmon, gmon, umon

rmon - integer; a variable that specifies which Ramtek monitor the red player's output will be displayed on

gmon - integer; a variable that specifies which Ramtek monitor the green player's output will be displayed on

umon - integer; a variable that specifies which Ramtek monitor the umpire's output will be displayed on.

COMMON /REPOS/ ad

ad - integer; a variable that is used as a subscript when repositioning units

COMMON /USERS/ reflg,grnflg

redflg - logical; a flag used when the red player reviews his unit positions. If the flag were not used, SUBROUTINE RPOSIT would call SUBROUTINE UNTPLT and try to redraw the units in their current positions. That would create an error condition.

grnflg - logical; a flag used when the green player reviews his unit positions. If the flag were not used, SUBROUTINE GPOSIT would call SUBROUTINE UNTPLT and try to redraw the units in their current positions. That would create an error condition.

C. THE GRAPHICS MODULES

Several subroutines were developed to perform certain functions. These in turn call various DI-3000 routines for generating the actual graphical objects for the bameboard. These routines are examined in this section.

Parameters are variables passed between routines in the call statement.

1. SUBROUTINE CHK1

File: HEX2.FOR

Purpose and Method:

This subroutine is used to compare the adjacent upper right hex to a given hex to see if a river flows between the two. If a river does flow between the two a graphics routine is executed that draws the river.

Parameters:

i, j - integers; coordinate pair for a two-dimensional matrix

Common Variables Referenced but Not Changed:

X1, Y1, iriver

Entries:

SUBROUTINE HEXX

2. <u>SUBROUTINE CHK2</u>

File: HEX2.FOR

Purpose and Method:

This subroutine is used to compare the adjacent lower right hex to a given hex to see if a river flows between the two. If a river does flow between the two a graphics routine is executed that draws the river.

Parameters:

i, j - integers; coordinate pair for a two-dimensional matrix

Common Variables Referenced but Not Changed:

X1, Y1, iriver

Entries:

SUBROUTINE HEXX

3. SUBROUTINE CHK3

File: HEX2.FOR

Purpose and Method:

This subroutine is used to compare the adjacent bottom hex to a given hex to see if a river flows between the two. If a river does flow between the two a graphics routine is executed that draws the river.

Parameters:

i, j - integers; coordinate pair for a two-dimensional matrix

Common Variables Referenced but Not Changed:

X1, Y1, iriver

Entries:

SUBROUTINE HEXX

4. SUBROUTINE RCHK1

File: HEX2.FOR

Purpose and Method:

This subroutine is used to compare the adjacent top right hex to a given hex to see if a road runs between the two. If a road does run between the two a graphics routine is executed that draws the road.

Parameters:

i, j - integers; coordinate pair for a two-dimensional matrix

Common Variables Referenced but Not Changed:

X1, Y1, iroad

Entries:

SUBROUTINE HEXX

5. SUBROUTINE RCHK2

File: HEX2.FOR

Purpose and Method:

This subroutine is used to compare the adjacent lower right hex to a given hex to see if a road runs between the two. If a road does run between the two a graphics routine is executed that draws the road.

Parameters:

i, j - integers; coordinate pair for a two-dimensional matrix

Common Variables Referenced but Not Changed:

X1, Y1, iroad

Entries:

SUBROUTINE HEXX

6. SUBROUTINE RCHK3

File: HEX2.FOR

Purpose and Method:

This subroutine is used to compare the adjacent bottom hex to a given hex to see if a road runs between the two. If a road does run between the two a graphics routine is executed that draws the road.

Parameters:

i, j - integers; coordinate pair for a two-dimensional matrix

Common Variables Referenced but Not Changed:

X1, Y1, iroad

Entries:

SUBROUTINE HEXX

7. <u>SUBROUTINE HEXX</u>

File: HEX1.FOR

Purpose and Method:

This subroutine is used by the umpire to assign and initialize Ramtek monitors, retrieve map files, and display new maps. COMEL uses this subroutine to read map data files and to draw new mapboards with roads and rivers.

Common Variables Reference but Not Changed:

ratio

Common Variables Changed:

iriver, iroad, x1, y1, rmon, gmon, umon, copyflag

Subprograms Called:

SUBROUTINE SETUP
SUBROUTINE RETRIEVE
SUBROUTINE CHK1
SUBROUTINE CHK2
SUBROUTINE CHK3
SUBROUTINE RCHK1
SUBROUTINE RCHK2
SUBROUTINE RCHK3
SUBROUTINE FILE

Data Files Accessed and File Specification:

11 - hex center coordinates	CENTER.DAT
11 - game board attributes	HEX.DAT
12 - river data file	VHEX.DAT
13 - road data file	RHEX.DAT

Entries: UMPIRE.FOR

8. SUBROUTINE RETRIEVE

File: RETFILE.FOR

Purpose and Method:

This subroutine is used to retrieve a previously saved map file and display it on the Ramtek display devices that have been initialized.

Common Variables Referenced but Not Changed:

umon, ratio, copyflag

Common Variables Changed:

mapname

Entries:

SUBROUTINE HEXX

9. SUBROUTINE FILE

File: RETFILE.FOR

Purpose and Method:

This subroutine is used to store a map file that has been drawn on the Ramtek display. The umpire assigns a name to the map file.

Common Variables Referenced but Not Changed:

ratio, umon

Entries:

SUBROUTINE HEXX

10. SUBROUTINE REPOS

File: REPO.FOR

Purpose and Method:

This subroutine is used to reposition units that are colocated with other units.

Parameters:

indx - integer; used as a counter for a do loop. It is used to count the number of times different units are encountered in the subroutine.

row - integer; passes the row value for the unit being checked

col - integer; passes the column value for the unit being checked

xinc - integer; x-increment for the x-coordinate of the unit being checked

yinc - integer; y-increment for the y-coordinate of the unit being checked

Common Variables Changed:

ad

Entries:

GEN6.FOR

11. <u>SUBROUTINE SETUP</u>

File: HEX1.FOR

Purpose and Method:

This subroutine is called to initialize the Ramtek display devices that the umpire has designated for use. It also defines the display area and defines some colors that are not default colors.

Parameters:

i - integer; used to indicate which Ramtek monitor is being initialized

Common Variables Changed:

ratio

Entries:

SUBROUTINE HEXX

12. SUBROUTINE RTURN/GTURN

File: RED1.FOR/GREEN1.FOR

Purpose and Method:

This subroutine offers a menu of possible actions, inputs the player's choice, calls the appropriate subroutine, and then loops back for another choice.

Initialize temporary values for equipment. Present player action menu.

Parameters:

stopat - real; scheduled time for end of turn

Common Variables Changed:

redflg

13. SUBROUTINE REDGOAL/GREENGOAL

File: RED1.FOR/GREEN1/FOR

Purpose and Method:

This routine allows the player to change the movement goal of a combat unit, displays a projected path, and allows the player to input an interim point to take another path.

Input unit name and check identity of the unit. Check communications link to see if change of orders can be sent to the unit. Display present goal of unit and ask player if he wants to change it. Ask for new goal, '0 0' to be input to stop a unit in its present position. Output new goal and proposed path. If the unit is a sealifted Marine force, the goal remains that of the Amphibious Task Force carrying the Marines.

Major Variables:

unitna - character*8; name of unit input by player

effc3 - real; effectiveness of communications link to JTFHQ

rng - integer; distance to hq

rn - real; random number

Common Variables Changed:

rgoal/ggoal

Common Variables Referenced but not Changed:

rmon/gmon/umon, redunt/grnunt, rrow/grow, rcol/gcol, rforce/gforce, rpathr/gpathr, rpathc/gpathc, rstart/gstart, rend/gend, rlenth/glenth, rseal/gseal, seed

Subprograms Called:

SUBROUTINE OPTIM
Real function C3EXT
Real function NCACON
System function RAN

Entries:

SUBROUTINE RTURN/GTURN

14. SUBROUTINE UNTPLT

File: UP.FOR

Purpose and Method:

This subroutine assigns an alphabetic character to each unit and displays the unit in its appropriate hex. It also draws a circle in the hex so the unit letter will be more visable on the hex.

Parameters:

i; a subscript used to identify units

hxx; a subscript used to designate the x-coordinate of a unit

hxy; a subscript used to designate the y-coordinate of a unit

segpr; a counter used to define a segment priority

segclr; a counter used to designate the color of the unit letter

Common Variables Referenced but Not Changed:

x1, y1

Entries:

SUBROUTINE RPOSIT, GPOSIT

15. SUBROUTINE ENDWAR

File: UMPFILEB.FOR

Purpose and Method:

This subroutine determines whether either side has won the war in a decisive victory. If not, it gives status to the Umpire and allows the Umpire to decide whether to artificially declare an end to the war with or without a marginal victory for one side.

Determine if either side controls the major objectives by checking the last owner of hexes listed in KEYA. Determine if either side controls the minor objectives in the same way, using hexes listed in KEYB. If the same side controls both major and minor objectives declare them the decisive victors and end the game. If one side controls the major objectives and the other side controls no minor objectives, the first side has a marginal victory. Allow the Umpire to decide whether to end the game. If one side controls all major objectives but the enemy controls any minor objectives, the outcome is indecisive. Allow the Umpire to decide whether to end the game and whether to declare the first side a winner. If neither side controls the major objectives, there is no

winner. Allow the Umpire the option of ending the game in a ceasefire.

Parameters:

over - logical; true if controller decides to end game or if one side has achieved a victory

redflg/grnflg - logical; true if subroutines RPOSIT/GPOSIT are called to prevent the subroutines from replotting the locations of the units

Other Major Variables:

row, col - integers; location of a key objective

major - character * 1; side which controls major objectives

minor - character * 1; side which controls minor objectives

winner - character * 5; side which is winning or ahead

Common Variables Referenced but Not Changed:

keya, keyb, lasown

Subprograms Called:

SUBROUTINE RPOSIT/GPOSIT
SUBROUTINE RLISTEN/GLISTEN

Entries: SUBROUTINE OPNCTL

16. SUBROUTINE OPNCTL

File: UMPFILE1.FOR

Purpose and Method:

This module provides timing and sequencing for the Operations Phase of the game.

Output Operations game header. Check whether starting a new game; get seed and call input subroutine or read blackboards. Display unit positions to set up map. Loop through turns. Input turn length and rules of engagement. If it is not the first turn, output intelligence. Output blackboards. Wait for the players to finish turns, checking periodically for messages or end of time limit. When time is up or both players are finished, continue with movement, combat, and end of war decision.

Major Variables:

hhmmss - character*8; time

first - integer; rules of engagement

ictl,igrn,ired,it,ishr,ismn - integers; action completion flags

durmin - real; length of turn in minutes

dummy,s,stemp - real; dummy variables

stopat - real; scheduled end of turn

over - logical; true if game is to end

Common Variables Changed:

conunt, redflg, grnflg, copyflag

Subprograms Called:

SUBROUTINE PAGE
SUBROUTINE DELAY
SUBROUTINE TIME

SUBROUTINE WRITIT
SUBROUTINE READIT
SUBROUTINE RMSG/GMSG
SUBROUTINE SYSOUT
SUBROUTINE SYSBRD
SUBROUTINE REDOUT/GRNOUT
SUBROUTINE RULES
SUBROUTINE OPINTEL
SUBROUTINE INPUT
SUBROUTINE RPOSIT/GPOSIT
SUBROUTINE RMOVE/GMOVE
SUBROUTINE DICTION
SUBROUTINE COMBAT
SUBROUTINE ENDWAR
SUBROUTINE REDBRD/GRNBRD
SUBROUTINE DELETE

Entries:

main program UMPFILE1.FOR

17. <u>SUBROUTINE DELETE</u>

File: GEN6.FOR

Purpose and Method:

This subroutine is used to delete the unit displays so the new positions can be plotted.

A call is made to JPURGE which deletes the entire display. COPYFLAG is used to recall the basic map and SUBROUTINE RPOSIT and SUBROUTINE GPOSIT are called to display the new unit locations.

Common Variables Referenced but Not Changed:

rmon, gmon, umon, copyflag, mapname, maxx, maxy

Subprograms Called:

SUBROUTINE RETRIEVE

Entries:

main program GEN6.FOR

18. <u>SUBROUTINE RPOSIT/GPOSIT</u>

File: GEN6.FOR

Purpose and Method:

This routine displays the unit name, location, goal, and combat points for each unit. It also passes the unit location and name to UP.FOR to plot the locations of the units on the graphics terminals. For each unit, write values presently in the common variables.

Parameters:

redflg/grnflg - logical; if true, subroutine untplt is not called

Major Variables:

maxx/maxy - integers; indicate the maximum number of red and green units

Common Variables Referenced but Not Changed:

rmon, gmon, umon

Common Variables Changed:

ad

Subprograms Called:

SUBROUTINE REPOS
SUBROUTINE BLINK
SUBROUTINE UNTPLT

Entries:

SUBROUTINE RTURN/GTURN
SUBROUTINE ENDWAR

19. <u>SUBROUTINE HILITE</u>

File: GEN2.FOR

Purpose and Method:

This subroutine is used to highlight hexes containing enemy units.

A DI-3000 routine is called that makes a specified color blink.

Parameters:

row/col - integers; subscripts that specify and x and y coordinate pair for a hex

Common Variables Referenced but Not Changed:

rmon, gmon, umon, x1, y1

Common Variables Changed:

conunt

Entries: SUBROUTINE ENEMY

20. SUBROUTINE ENEMY

File: GEN2.FOR

Purpose and Method:

This subroutine surveys HEX(row,col) and adjacent hexes for enemy units.

Identify adjacent hexes and initialize logical to false. Survey hexes; if owner is other side or both, set logical.

Parameters:

row,col - integers; center hex

side - character * 1; side doing check

cnflct - logical; true if other side is present

Other Major Variables:

iu, id, ius, ids, jl, jr - integers; row or column of adjacent hexes

Common Variables Referenced but Not Changed:

owner

Subprograms Called:

SUBROUTINE HILITE

Entries:

SUBROUTINE FIGHT
SUBROUTINE MVMENT
SUBROUTINE RMOVE/GMOVE

21. SUBROUTINE RTURN/GTURN

File: RED1.FOR/GREEN1.FOR

Purpose and Method:

This subroutine offers a menu of possible actions, inputs the player's choice, calls the appropriate subroutine, and then loops back for another choice.

initialize temporary values for equipment. Present player action menu.

Parameters:

stopat - real; scheduled time for end of turn

Major Variables:

choice - character*1; menu selection

airlft - logical; true if an airdrop has already been requested for this turn

Common Variables Referenced but Not Changed:

redflg/grnflg

Common Variables Changed:

rrecce/grecce

Subprograms Called:

SUBROUTINE REDGOAL/GRNGOAL
SUBROUTINE RCOMM/GCOMM
SUBROUTINE RRECON/GRECON
SUBROUTINE RCACAS/GCACAS
SUBROUTINE RPOSIT/GPOSIT
SUBROUTINE RSEALIFT/GSEALIFT
SUBROUTINE RAIRLIFT/GAIRLIFT
SUBROUTINE RATKSAT/GATKSAT
SUBROUTINE RTERAIN/GTERAIN

Entries:

SUBROUTINE REDOPS/GRNOPS

22. SUBROUTINE REDGOAL/GRNGOAL

File: RED1.FOR/GREEN1.FOR

Purpose and Method:

This routine allows the player to change the movement goal of a combat unit, displays a projected path on the Ramtek display and on the terminal, and allows the player to input an intermediate point to take another path.

Input unit name and check identity of the unit. Check communications link to see if change of orders can be sent to the unit. Display present goal of unit and ask player if he wants to change it. As for new goal, '0 0' to be input to stop a unit in its present position. Output new goal and proposed path. If the unit is a sealifted Marine force, the goal remains that of the Amphibious Task Force carrying the Marines.

Major Variables:

unitna - character*8; name of unit imput by player

effc3 - real; effectiveness of comm link to JTFHQ

rng - integer; distance to hq

rn - real; random number

Common Variables Changed:

rgoal/goal

Common Variables Referenced but Not Changed:

redunt/grnunt, prathr/gpathr, rpathc/gpathc, rrow/grow, rcol/gcol, rforce/gforce, rstart/gstart, rend/gend, rlenth/glenth, rseal/gseal, seed, rmon, gmon, umon

Subprograms Called:

SUBROUTINE OPTIM
Real Function C3EXT
Real Function NCACON
System Function RAN

Entries:

SUBROUTINE RTURN/GTURN

D. DI-3000 CALLS

The DI-3000 routines that are used throughout COMEL are explained to provide the user with a general understanding of what each command accomplishes. For a more detailed explanation of these variables the user should consult the DI-3000 User's Guide. [Ref. 2]

1. JASPEK (DSPDEV,RATIO)

This routine returns the display surface aspect ratio of a specific graphics display device. The aspect ratio defines the shape of the visible region of a display device.

DSPDEV defines an initialized graphics display device.

RATIO is the aspect ratio (height/width) of DSPDEV.

2. JBEGIN ()
   JEND ()

JBEGIN initializes DI-3000 and guarantees that DI-3000 is in the system initialization state with all default parameters, attribute, and mode settings established.

JEND terminates DI-3000 and insures that all graphics output is complete to all selected display devices. Any initialized and selected display devices are deselected and terminated.

3. JBGBAT (LEVEL)

This routine is used to begin a batch of updates that enables the DI-3000 to process a large number of picture changes without displaying each picture change. The composite picture is the only one displayed so the amount of time utilized in calls to the display device is reduced.

LEVEL sets the level of the update.

4. JCIRCL (XO, YO, ZO, RADIUS, NSEG)

JCIRCL is used to output a circle on a projection plane parallel to the xy-plane of the world coordinate system (monitor surface).

XO, YO, and ZO designate the center of the circle in world coordinates.

RADIUS specifies the radius of the circle in world coordinates.

NSEG specifies the number of line segments to be used to draw the circle. If NSEG is less than three, JCIRCL will choose a number of segments relative to the length of the radius to make the circle as smooth as possible.

5. JCOLOR (CVALUE)

JCOLOR specifies the current color to be used when displaying objects on the display device.

CVALUE specifies one of the colors available for the display. Colors 0 thru 8 are preset and colors 9 thru 32,767 are defined by the user.

6. JCOTBL (DSPDEV, COUNT, INDEXS, HUES, SATS, LIGHTS)

This feature allows the user to download portions of the color table on a specified display device.

DSPDEV specifies a particular display device.

COUNT sets the number of elements of the device level color table that are to be defined.

INDEXS lists the array of color table indixes to be defined. Since colors 0 thru 8 are reserved (see JCOLOR), the user can specify color numbers 9 thru 32,767.

HUES is an array of values that make it possible to classify color as red, green, blue or intermediates of the three primary colors. Hue is defined as an angle in degrees. A value of 0 indicates pure blue, 120 indicates pure red, and 240 indicates pure green. Intermediates are specified by indicating a value between the primaries; e.g., yellow would be 180 (between red and green). SATS is an array that determines the degree to which a color differs from a gray of the same lightness. A value of 0 indicates pure gray and a value of 32,767 indicates a pure color with no gray.

LIGHTS determines the amount of lightness assigned to each color. A value of 0 indicates that the color is black (no light) through various grays to 32,767 or white (maximum light).

7. <u>JDINIT (DEVICE)</u>
   <u>JDEVON (DEVICE)</u>
   <u>JDEVOF (DEVICE)</u>
   <u>JDEND (DEVICE)</u>

These four commands are used together. DEVICE specifies which display device is to be addressed. JDINIT initializes the display device and must be called before any output can be sent to that device. JDEVON turns the device on, JDEVOF turns the display device off, and JDEND deselects the device and terminates it.

8. JDRAW (X,Y)

This routine is used to draw a line from the current position to the specified world coordinate position. The new position is determined from the arguments associated with JDRAW.

X specifies the x-coordinate of the new location.

Y specifies the y-coordinate of the new location.

9. JENBAT ()

JENBAT is used to signify the end of the batch picture processing (see JBGBAT).

10. JESCAP (CODE,NINTEG,NREAL,ILIST,RLIST)

There are several escape CODES that can be called in the DI-3000 library that provide an access mechanism for a device independent DI-3000 application program to utilize special hardware capabilities of the selected display surfaces. Three calls to JESCAP with different values for the CODE were made in the graphics routines. The values associated with them will be explained after each of the arguments.

CODE specifies which function code is being accessed.

9408 - indicates that a particular color in the color table is to blink.

9424 - used to write a properly formatted section file (pixel file) to a Ramtek.

9425 - used to read a window of pixels from the Ramtek and write a properly formatted section file. The section file must have already been created using either KSCRET or KSOPEN.

NINTEG is the number of integer parameters to the escape function. It is the dimension of ILIST.

9408 - the number of integers used in this call is 2.

9424 - the number of integers used in this call is either 1 or 2.

9425 - the number of integers used in this call is either 1 or 2.

NREAL sets the number of virtual coordinate parameters passed to the escape function and is the dimension of RLIST.

9408 - the number of reals used in this call is 0.

9424 - the number of reals used in this call is either 0 or 2.

9425 - the number of reals used in this call is either 0 or 4.

ILIST is the array of integer parameters passed to the escape function.

9408 - the first array element specifies the color table index. The second array element turns off the blink (0) or turns on the blink (1).

9424 - the first array element designates the first address of the section file as specified by KSOPEN. The second element, if used, specifies a bit plane erase mask.

9425 - the first array element is the address of the section file as returned by KSOPEN/KSCRET. The second element, if used, specifies a bit plane read mask.

RLIST is the array of real parameters passed to the escape function.

9408 - there are no real array elements for this call.

9424 - if used, the two array elements specify the x- and y-coordinates of the lower left corner of the window into which the pixels are written. If not specified, the coordinates that are used are the same ones that are used to read the section file.

9425 - if used, the two elements designate the x- and y-coordinates of the lower left corner of the window into which pixels are written.

11. <u>JFILES (CODE, DSPDEV, FILSPC)</u>

JFILES overrides the default specifications for the logical unit number of a DI-3000 internal file.

CODE indicates the DI-3000 internal code to be changed. Four internal files can be changed. They are the error file, debug file, graphics output file, and graphics input file.

DSPDEV is an uninitialized display device that will later be initialized.

FILSPC is the new file specification for the internal file and is usually a FOTRAN logical unit number.

12. <u>JFONT (CVALUE)</u>

JFONT sets the current character font primitive attributes.

CVALUE defines the character font of subsequent primitives within a currently open segment.

13. <u>JHTEXT (NCHARS, STRING)</u>

JHTEXT outputs a graphic arts quality test string as an output primitive. It is the highest quality text output. Lower quality text output is available by calling J1TEXT, J2TEXT, and J3TEXT.

NCHARS is an integer value that specifies the number of characters to be written on the display device.

STRING is the literal string to be output to the display device.

14. JINTEN (CVALUE)

JINTEN sets the current intensity of the primitive attribute.

CVALUE defines the level of intensity with values ranging from 0 thru 32,767.

15. JLWIDE (CVALUE)

JLWIDE sets the current line width primitive attribute.

CVALUE specifies the line width with the range going from 0 to 32,767.

16. JMOVE (X,Y)

JMOVE moves the cursor from its present location to a new location without drawing any lines on the display device using absolute world coordinates.

X defines the x-coordinate of the new position.

Y defines the y-coordinate of the new position.

17. JPEDGE (CVALUE)

JPEDGE defines the current polygon edge style. The polygon edge style indicates the method of forming the border of a visible polygon.

CVALUE specifies the polygon edge style within the currently opened segment. It can have values from 0 thru 32,767. Even values indicate that the edge style is solid. Odd values indicate that the edge style is to have the same attributes as the interior of the polygon (see JPINTR) with no seperate border.

18. <u>JPIDEX (CCOLOR,CINTEN)</u>

JPIDEX sets the current interior color and intensity attributes.

CCOLOR sets the interior color of subsequent polygons within the currently opened segment. The range for CCOLOR is 0 to 32,767.

CINTEN sets the intensity level of subsequent polygons within the currently opened segment. The range for CINTEN is 0 to 32,767.

19. <u>JPINTR (CVALUE)</u>

JPINTR sets the current polygon interior style attributes which determines the method of filling the interior of a polygon.

CVALUE specifies which style of interior fill is to be used. 0 indicates that the polygon is not to be filled. 1 indicates that the polygon is to be solid filled with the color specified by JPIDEX. Other values (2 thru 32,767) are device dependent values.

20. <u>JRPLGN (DX,DY,N)</u>

JRPLGN outputs a polygon in 2-D relative world coordinates. A polygon is created by an invisible move to the first point, line drawings to the subsequent points in the array, with a final draw from the Nth point in the array back to the first point. This option allows the polygon's interior to be filled.

DX specifies the x-coordinate of the center of the polygon.

DY specifies the y-coordinate of the center of the polygon.

N defines the number of points in the polygon array.

21. <u>JPURGE (NAME)</u>

JPURGE deletes a retained segment and releases the segment storage space allocated to NAME.

NAME specifies which retained segment is to be deleted. If a retained segment NAME does not exist, an error message will be printed.

22. <u>JROPEN (NAME)</u>

JROPEN opens a retained segment.

NAME is the name of the retained segment and is an integer value. It can have a value from 1 to 32,000.

23. <u>JSETDB (LEVEL)</u>

JSETDB sets the DI-3000 graphics debugging level. By changing the debugging level the verbosity of the traceback statements will be changed.

LEVEL specifies the level of verbosity for JSETDB. The levels range from 0 (no comments) to 7 (full DI-3000 traceback).

24. <u>JSGPRI (NAME, SEGPRI)</u>

JSGPRI controls the segment priority of a retained segment and determines the order (from front to rear) in

which the retained segments are displayed on an initialized display device.

NAME is an integer value that specifies whose segment priority is to be changed.

SEGPRI specifies the priority of segment NAME. A higher value segment will be displayed over a lower value segment. A value of 0 indicates that retained segments have no priority. Values ranging from 1 to 32,767 specify a segment priority.

25. JVSPAC (VXMIN, VXMAX, VYMIN, VYMAX).

JVSPAC redefines the dimensions of the virtual coordinate system and the mapping from the virtual coordinate system onto the visible region of all selected display devices. When JBEGIN is called, the default virtual coordinates of the display device are VXMIN = -1.; VXMAX = 1.0; VYMIN = -1.0; and VYMAX = 1.0. JVSPAC changes the boundaries to the values specified.

VXMIN sets the minimum value for the virtual x-coordinate boundary.

VXMAX sets the maximum value for the virtual x-coordinate boundary.

VYMIN sets the minimum value for the virtual y-coordinate boundary.

VYMAX sets the maximum value for the virtual y-coordinate boundary.

26. JWINDO (UMIN, UMAX, VMIN, VMAX)

JWINDO defines the boundaries of a window in the real world coordinate system. When JBEGIN is called the world coordinate boundary limits are XMIN = -1.0; XMAX = 1.0; YMIN = -1.0; and YMAX = 1.0. JWINDO lets the user define the minimum and maximum values of the display area.

UMIN is a real value that specifies the minimum value of the horizontal axis.

UMAX is a real value that specifies the maximum value of the horizontal axis.

VMIN is a real value that specifies the minimum value of the vertical axis.

VMAX is a real value that specifies the maximum value of the vertical axis.

27. KSBYTE24 (DSPDEV, VXLL, VYLL, VXUR, VYUR)

KSBYTE24 is used to get the number of bytes of pixels used to display an image in a virtual screen window. This version of KSBYTE also includes overhead associated with calling JESCAP 9424 and JESCAP 9425.

DSPDEV specifies the device to be examined. It is a value that is entered by the user.

VXLL and VYLL specify the x and y coordinates respectively for the lower left corner of the virtual coordinates of the display area. They are values that are specified by the user.

VXUR and VYUR specify the x and y coordinates respectively for the upper right corner of the virtual coordinates of the display area. They are values that are specified by the user.

28. KSCRET (FILENAME, ISIZE, LUN, ICHANNEL, IADDR, IERR)

KSCRET is used to create a section file of a specified size and is mapped into the user's virtual address space (user directory).

FILENAME is the name of the section file to be created. It is a character string that is specified by the user.

ISIZE is the size of the file in bytes. It is a value that is determined by calling KSBYTE or KSBYTE24. The created file size is rounded up to an integral number of blocks (512 bytes = 1 block).

LUN is the logical unit number on which the file is being created (mass storage device such as a disk).

ICHANNEL is the channel number that the operating system assigns to the mass storage device for data transfer.

IADDR is an array of two elements that specify the starting and ending addresses of the section file where it now exists in the segment storage area. Both are -1 if the mapping failed.

IERR is a flag value that is passed if the file could not be opened. KSCRET will display an error message if IERR returns a value other than 0.

29. KSOPEN (FILENAME, LUN, ICHAN, PADDR, IERR)

KSOPEN is used to open a section file and map the file into the space that has been defined for the display device.

FILENAME is the name of a previously stored section file. It is specified by the user.

LUN is the logical unit number that the operating system assigns to the mass storage device for data transfer.

ICHAN is the channel number that the operating system assigns to the mass storage device for data transfer.

PADDR is a two dimensional array that indicates the opening and ending address of the section file in the segment storage area. Both are -1 if the mapping failed.

IERR is an error flag that is returned if the file could not be opened. The flag would cause KSOPEN to display an error message on the terminal.

30. KTOHSL (R, G, B, H, S, L, NDAC)

KTOHSL is used to define colors using their red, green, and blue values. The routine returns values for hues, saturations, and lights which are passed to JCOTBL.

R is the brightness of red to be used. 0 is no red and 15 is the maximum amount of red. This value is specified by the user.

G is the brightness of green to be used. The range of values is the same as for red. This value is specified by the user.

B is the brightness of blue to be used. The range of values is the same as for red. This value is specified by the user.

H is the value of the hue (see JCOTBL for a definition of hue). It is a value returned by KTOHSL as a result of the values entered for R, G, and B. That value is then passed to JCOTBL.

S is the value of the saturation level (see JCOTBL for a definition of saturation). The other attributes are similiar to H.

L is the value of the light level (see JCOTBL for a definition of light). The other attributes are similar to H.

NDAC specifies the number of bits for each color for the color look-up table. The value can either be 4 or 8. If the value is 4 the range for R, G, and B is from 0 to 15. If the value is 8 the range for R, G, and B is from 0 to 255.

E. COMPLETE MAINTENANCE MANUAL

This supplementary maintenance manual has been merged with the original manual and is in the WAR Lab computer. It is accessible from the COMEL directory.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22314

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

3. Professor M. Sovereign, Code 74 2
Chairman C3 Academic Group
Naval Postgraduate School
Monterey, California 93943

4. CDR G. Porter, Code 55Pt 4
Director, Gaming WAR Lab
Naval Postgraduate School
Monterey, California 93943

5. CPT Martin R. Conrad, Jr. 1
219 North Esplanade
Leavenworth, Kansas 66048

6. Lt Col J. Malokas, Code 39 2
C3 Curricular Officer
Naval Postgraduate School
Monterey, California 93943

7. U. S. Army War College 1
ATTN: MAJ Howard Yellen, USA
Carlisle Barracks, Pennsylvania 17013

8. Dr. A. M. Zied, Code 74Zi 1
Technical Director, WAR Lab
Naval Postgraduate School
Monterey, California 93943

9. Joint Telecommunications Systems/ 1
Staff Officers Course (TSSOC)
ATTN: Maj Perrine
Keesler AFB, Mississippi 39534

206064

Thesis
C7416 Conrad
c.1 Computer graphics
enchantment for COMEL.