AD-A083 911 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 15/7
STAG: A TWO PERSON SIMULATED TACTICAL AIR WAR GAME.(U)
MAR 80 J M FOLEY
UNCLASSIFIED AFIT/GST/OS/80M-2 NL

1 OF 2
AD A083911

AFIT/GST/OS/80M-2

LEVEL II

ADA083911

STAG: A TWO PERSON SIMULATED

TACTICAL AIR WAR GAME

THESIS

AFIT/GST/OS/80M-2 John M. Foley
Capt USAF

80 5 6 065

AFIT/GST/OS/80M-2

STAG: A TWO PERSON SIMULATED TACTICAL AIR WAR GAME

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
John M. Foley, B.A.
Capt USAF
Graduate Strategic and Tactical Sciences

March 1980

Preface

While attending the Combined Air Warfare Course at Maxwell AFB, Alabama, I had the opportunity to participate in the Theater War Exercise (TWX), a theater-level war game. The experience was enlightening and rewarding. However, I was nearly overwhelmed by the tremendous amount of data involved in playing the game. Consequently, I decided to develop a war game with a limited number of factors so that it would be easier for players to determine the impact of their strategies on the outcome of the war.

I wish to thank Lt Col James Havey, my faculty advisor, for his assistance and sound advice throughout this effort. In addition thanks are extended to my readers, Lt Col Tom Clark and Capt Dan Fox, for their time and encouragement; and to my typist, Cindi Prater, for her tireless efforts to produce a quality product.

Finally, I wish to thank my wife, Nani, whose usually gentle prodding ensured the successful completion of this research.

John M. Foley

TABLE OF CONTENTS

		Page
Preface		ii
List of Figures		v
List of Tables		vi
Abstract		vii
I	INTRODUCTION	1
	Model	4
	Simulation	5
	Game	5
II	GENERAL DESCRIPTION	7
	Assumptions	12
	Limitations	13
III	AIR WARFARE	14
	Air Base Attack	15
	Interdiction	20
	Defense Suppression	21
	Escort	22
	Air Defense	22
	Close Air Support	23
	Reconnaissance	24
	Reinforcements and Logistics	25
	Aircraft Losses	27
IV	GROUND WARFARE	32
	Introduction	32
	Composition and Control	32
	Movement of the FEBA	33
	Troop Casualties	34

		Page
V	VERIFICATION AND VALIDATION	37
	Rationalism	38
	Empiricism	39
	Pragmatism	39
	Utilitarianism	40
	Verification Tests	40
	Validation Test	43
VI	SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS	46
Appendix A		50
	User's Guide	50
	Annex	56
Appendix B		60
	Analytical Formulations	60
Appendix C		71
	Source Listing of Program	71
Bibliography		105
Vita		107

LIST OF FIGURES

Figure — Page

1. Theater of Operations 8
2. Air Mission Interactions 10
3. Attack Mission Flow 18
4. Ground Defenses 28
5. Engagement Ratio 31
6. Relationship of Ground Casualties to Force Ratio... 36

B-1. Effect of Constants on FEBA Movement 66

B-2. Cumulative FEBA Movement 67

LIST OF TABLES

Table		Page
I	Blue Aircraft Capabilities	16
II	Red Aircraft Capabilities	16
III	Daily Aircraft Reinforcements	25
IV	Blue Logistics Data	26
V	Red Logistics Data	26
VI	Results of Multiple Mode Experiment	42
VII	Results of Validation Experiment	43
A-I	CAS Destructive Indexes	55
A-II	ABA Destructive Indexes	55

Abstract

The model developed in this study is a highly aggregated theater-level game comprised of interaction equations which utilize the allocation of aircraft to various missions on a daily basis to obtain the outcome of an offensive versus defensive systems engagement. The simulation which supports the model consists of an interactive air portion and a parametric ground portion. The theater of operations consists of two sides with their respective air and ground forces. While the model produces credible outcomes, the main objective is to reflect the effect of strategy and employment tactics on the outcome of the battle.

The model is designed to provide individuals an opportunity to plan and conduct an air war and to test various air employment concepts. Existing war games are quite large and contain so many factors that the main effects of a player's employment decisions are confounded by the interactive effects of the factors. The war game described in this report has a limited number of factors so that it is easier for players to determine the main effects of their strategies. Included in the appendicies are a user's guide, analytical formulations, and a source listing of the program.

I. Introduction

During the course of a tactical air campaign, the commanders on each side are faced with many decisions which affect the outcome of the campaign. They make decisions such as how many sorties should be flown in offensive, defensive, or support roles; specific targets to be hit; mission profiles; and the mix of aircraft to be sent against each target. Two of the most important and basic decisions in a tactical air war are the apportionment of sorties among the various air tasks and the allocation of aircraft to be sent against each target.

The formidable task of apportioning sorties among offensive, defensive, or support roles and of allocating aircraft within the different air roles (air base attack, close air support, etc.) in a multi-strike campaign can be simulated using a computerized war game. A theater air commander must determine the appropriate mix of forces in various roles to achieve the objectives of tactical air power (Ref 2:16). The decision process employed by a commander can be characterized as a two-sided war game in which the successive decisions which are made each day are based upon the resources available and the status of enemy forces (Ref 3:4).

The magnitude of the problem confronting the tactical air commander is considerable. He will be dealing with thousands of sorties involving a dozen or more different types of aircraft. Several detailed simulation models have been developed to study the employment of tactical air forces, and these models have continued to increase in size and complexity in an attempt to approach an exact model of the real life situation. The problem with many of these models is their enormous size. The data bases are huge, and the computer storage space required to run these models severely limit where they can be operated. Model verification poses another major problem. For example the IDA TACWAR model (a comprehensive theater-level model developed for the Joint Chiefs of Staff) requires 10,000 data items to be input for model operation. Inconsistencies in input data are difficult to detect and sometimes remain undetected for a number of runs. In one case the user forgot to provide any ammunition to a portion of the ground force. Since that portion was expected to be quickly defeated, the lack of ammunition was undetected for sometime (Ref 11:V-20).

Oı the 152 listings described in the most recent edition of the Catalog of War Gaming and Military Simulation Models of the Studies, Analysis and Gaming Agency (SAGA), less than ten used human participants directly (Ref 1:111). The vast majority are machine simulations used mainly for analysis, diagnosis, and operational applications. The war games that do exist are large and detailed making it difficult for the participants to observe the full impact of their strategies.

Consequently, there is a need for a game which allows students of tactical air operations to create their own battles and to test their own strategies against those of an opponent with a program which is simple to use and inexpensive to run on a computer. Such a war game would deal with the apportionment and allocation decisions and be used as a primer on tactical air operations preparing students for participation in any one of the more complex war games used throughout the DOD (Ref 22:3). The level of detail in the game would be such that participants could readily observe the impact of their allocation decisions, note where they had made mistakes, and formulate new strategies.

The objectives of this research are twofold: to develop such an informal, two-person war game in which the players make decisions and supply input data as the game progresses from one day to the next; and to fully document the game for users and analysts. Since it involves two sides, it is a dynamic game in which the more important aspects of tactical air operations are not solely dependent on predetermined constant data. The primary purpose of the simulation is to provide individuals an opportunity to plan and conduct an air war and to test various air employment concepts. There are opportunities to capitalize on the mistakes of an opponent, and at the same time, to adjust one's own strategy to accomodate the lessons learned from the previous day's operations. An element of uncertainty is introduced into the game by having two competing sides with differing force structures and operational concepts (Ref 22:11).

Throughout this discussion reference is made to models, simulations, and games. Although these terms are often used interchangeably, they actually have quite different meanings. Some basic definitions are presented to clarify these differences.

Model

As defined by Brewer and Shubik, "a model is a representation of an entity or situation by something else that has the relevant features or properties of the original." There are five basic types: verbal models, analytic models, diagrammatic models, analog simulations, and digital simulations (Ref 5:10). A computerized model contains the rules, methodology, techniques, procedures, and logic required to approximate reality.

SIMULATION

The use of a model under specific conditions to represent and study the behavior of events, processes, or systems is simulation.

GAME

A game employs human beings acting as themselves or playing roles in an environment that is either actual or simulated. As defined by the Department of Defense, a war game is "a simulated military operation involving two or more opposing forces and using rules, data, and procedures designed to depict an actual or hypothetical real life situation" (Ref 5:8). War games are commonly used for three main purposes today: training, operational experience, and research. The game developed in this research is a training game designed to provide the participants with decision making opportunities similar to those they might experience in combat. It is the least complex of the three types (training, operational, or research) of games. An operational game employs current equipment, tactics, and strategy to test operational plans. Games designed for research are the most complex and are used to study future tactics or force structures.

The remainder of this thesis will pursue the objectives previously outlined. The second chapter will present a general description of the game developed in this research. Chapters III and IV describe the simulation for the air and ground portions of the model. The fifth chapter outlines the verification and validation of the model, while the final chapter contains a summary, conclusions, and recommendations. Included in the appendices are a user's guide detailing the specific directions for playing the game, the formal mathematical foundations used in the algorithms, and a source listing of the computer program defining the variables and arrays.

II. General Description

Included in this chapter is a brief description of the game itself, the eight possible missions which are simulated, and the assumptions and constraints inherent in the treatment of functional areas by the model.

The model STAG - Simulated Tactical Air War Game, is a highly aggregated game comprised of interaction equations which utilize the allocation of aircraft to various missions to obtain the outcome of an offensive vs. defensive systems engagement. The theater of operations involves two sides with their respective air and ground forces. Figure 1 provides a representation of the forces and the types of interactions included in this model.

Each side has three forward operating bases which are vulnerable to attack by the opponent. Aircraft replacements and supplies are generated from a sanctuary base located in the rear of the sector. The sanctuary base is assumed to be invulnerable to attack.

The theater is divided into two fairly well defined territories by a line called the forward edge of the battle area (FEBA). Since only one sector of the ground war is modeled in STAG, the FEBA moves as a unit and may be viewed as the average movement of the entire theater front.

The ground forces are defined in terms of homogeneous divisions with no distinction made between armored, infantry, or tank divisions. The primary purpose of the red side is

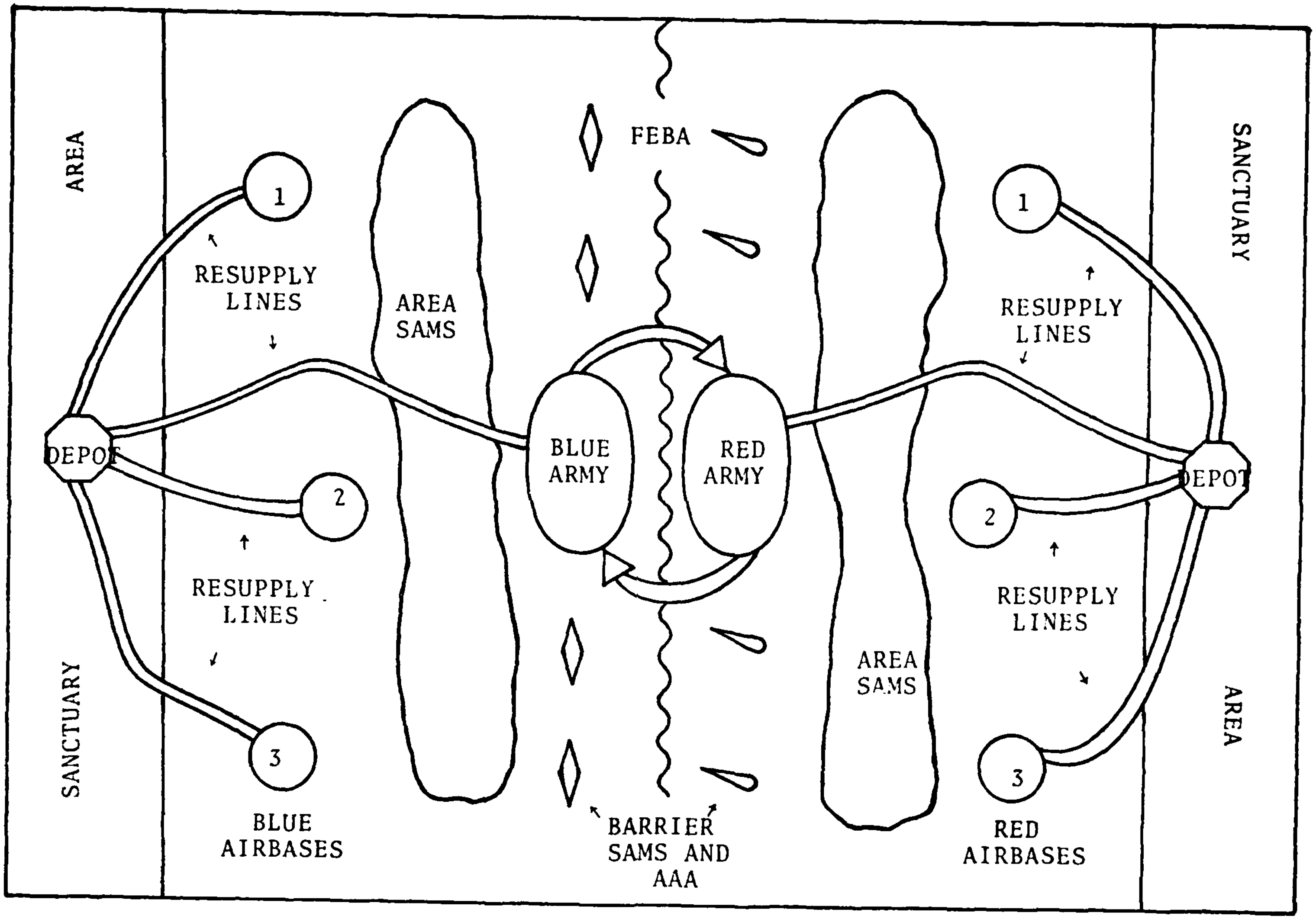

FIGURE 1. DEPICTION OF THEATER MODEL

to occupy territory, while blue's goal is to slow the rate of movement of the FEBA as much as possible. The direction and rate of movement of the FEBA depends upon the relative strength of the opposing forces. The supply system may be viewed as a pipeline running from the sanctuary area to the divisions positioned along the FEBA. Logistics is handled by using one aggregated type of supply which is designated as a "spare". Interdicting the supply system will reduce the number of spares delivered that day. The ground forces defend against enemy aircraft be means of anti-aircraft fire and surface-to-air missiles (SAMs).

The air forces consist of three general types: a multi-purpose fighter and two special mission aircraft. The multi-purpose fighter is capable of performing most of the various missions while the special aircraft are limited to a particular role (bomber or attack). The players on each side can allocate their aircraft among eight air missions provided for in the model: (1) airbase attack (ABA); (2) reconnaissance (RECCE); (3) interdiction (INTD); (4) combat air patrol (CAP); (5) close air support (CAS); (6) air defense (AIRDEF); (7) defense suppression (DEFSP); and (8) escort (ESCORT).

In the airbase attack mission (Ref Figure 2), offensive strikes are aimed at enemy air bases to destroy enemy aircraft on the ground, petroleum (POL), munitions, and to disrupt operations of the airbase.

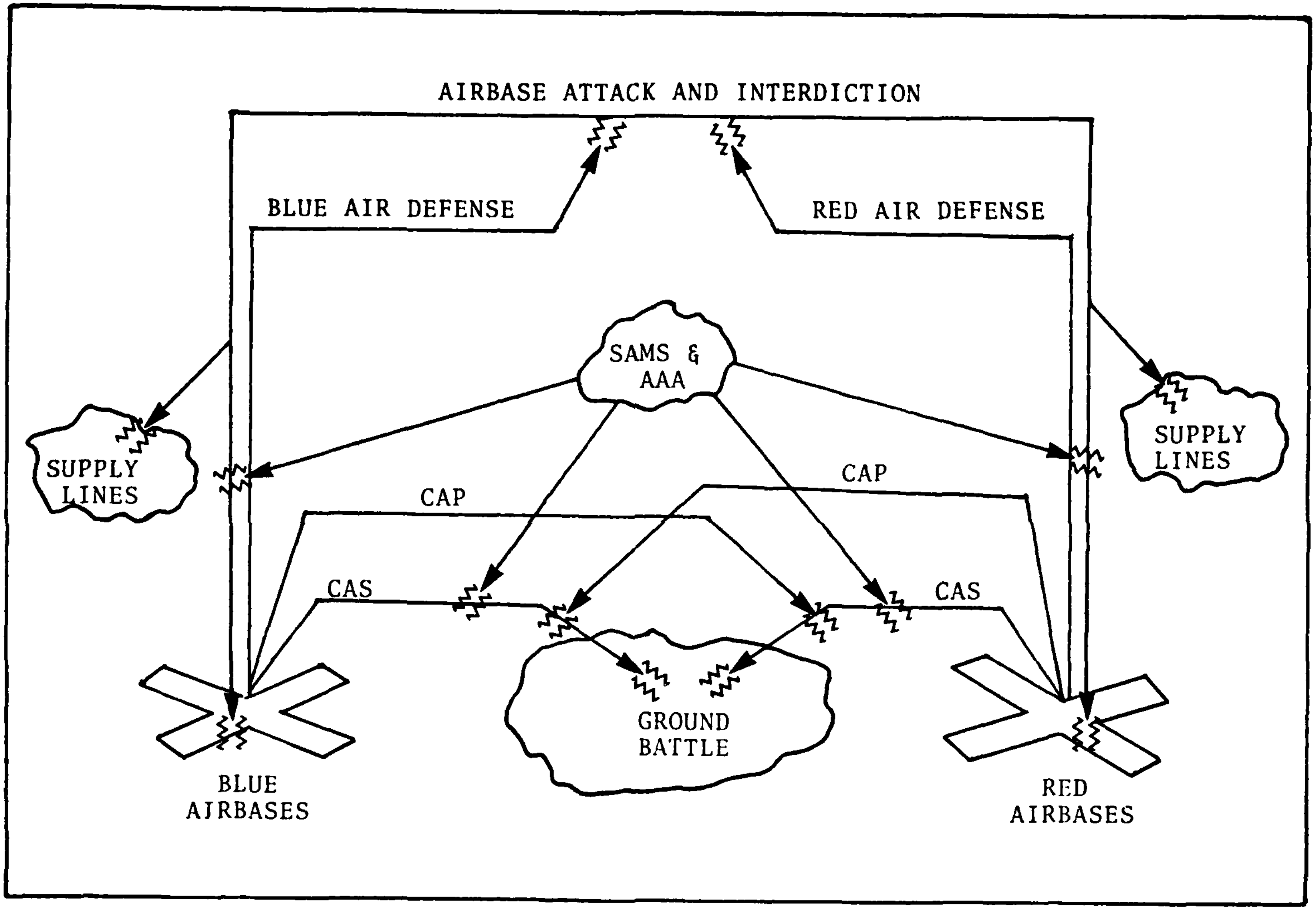

FIGURE 2. THEATER AIR OPERATIONS

The reconnaissance mission improves the accuracy of information about enemy airfields and ground forces. Flying reconnaissance missions against a target will give you the present status of supplies and logistics located at the target.

The purpose of the interdiction mission is to damage, destroy, or delay logistics support for enemy ground units engaged in battle. Successful interdiction missions will create a delay in arrival of resupplies and will also reduce their quantity.

Combat air patrol missions attempt to gain and maintain air superiority over the main battle area. These missions will tend to increase the effectiveness and reduce the losses of close air support.

The CAS missions attack enemy ground units engaged in combat with friendly forces. They have two principal effects. First, they produce casualties among ground units; and second they influence the movement of the FEBA by causing casualties, disrupting coordination, and slowing troop movement.

The air defense missions are aircraft on alert at designated bases and are used to protect that airbase from attack. In addtion, they protect territory behind the forward defenses from enemy aircraft which have penetrated the missile defense belt.

Defense Suppression missions are designed to destroy or suppress enemy ground-to-air defenses by clearing corridors for subsequent penetration by aircraft on interdiction or airbase attack missions.

Escort missions accompany primary mission aircraft, such as airbase attack, and engage enemy interceptors. These missions are part of a "mission package" concept used by the United States Air Force (USAF) in which attack aircraft are accompanied by properly configured escort and defense suppression aircraft. These escorts will reduce the losses to the primary mission aircraft from enemy interceptors and ground defenses. However, the "cost" is in terms of what one must sacrifice to provide the escort package.

ASSUMPTIONS

The following assumptions are made in this model:

(a) The conflict is a conventional war. Nuclear or chemical weapons are not modeled.

(b) Intangible quantities such as leadership and training are equal for each side and are not treated.

(c) Weather is not treated.

(d) Different types of munitions are not considered.

(e) Command, control, and communications (C^3) are not a factor.

(f) Air refueling, search and rescue, and aircrew training are not modeled.

(g) No distinction is made between a day light cycle or a nighttime cycle.

LIMITATIONS

This game is intended to be an educational tool, and is not meant to give real world results. Data contained in the model are either fictitious or from unclassified models presently in use. Many of the details of war gaming are deliberately suppressed in STAG since it is not meant to be an explicit representation of real-world events. However, the game should allow players to gain insights into the critical elements which must be considered in an effective air campaign.

III. AIR WARFARE

A standard force built into the model provides each side with three different types of aircraft. Each type possesses its own performance characteristics and capabilities reflected in a destructive index for that type. In general, the blue side possesses more effective aircraft systems which can counter the numerical advantage of red's ground forces if they are employed effectively. An option in the computer program also allows players to input their own force list (numbers of aircraft, not types) in lieu of the standard force.

The three types of aircraft are distributed at three airbases lying 200 kilometers behind the forward edge of the battle area (FEBA). Prior to the start of each day's activities, players are given the opportunity to move aircraft from one airbase to another. Aircraft beddown is then displayed prior to sortie allocation.

One task confronting the players is to provide support for the ground forces without entirely sacrificing the capability to destroy the enemy's air forces. Each side should base its allocation decisions on its own objectives, previous events, friendly ground forces' requirements, and its own estimate of the situation.

Aircraft may not be allocated to missions for which they are not suited. Tables 1 and 2 list the mission capabilities for each aircraft for each side. The air defense of a particular base must be performed by aircraft located at the base. For example if blue has 50 F-4 aircraft at base 2, no more than 50 F-4 aircraft can be allocated to the air defense of base 2. In order to allocate more than 50 aircraft for air defense alert at base 2, the player would have to move aircraft from base 1 or 3 to base 2 at the start of the day. Once both sides have finished allocating their sorties, the program calculates losses and provides a quantitative assessment of the air missions.

The missions of tactical air power are air base attack, interdiction, close air support, air defense (including combat air patrol), and special support missions including escort, defense suppression, and reconnaissance. The following paragraphs describe the operational considerations of the various air missions and the manner in which the computerized model simulates the interactions of these air missions. While it is desirable that the model outcomes be credible, primary emphasis is placed on reflecting the effect of strategy and employment tactics on the outcome of the battle.

AIR BASE ATTACK

Traditionally, air base attack (ABA) sorties have been one of the most effective methods of countering enemy air forces; they impact upon an opponent's air field by

TABLE 1

BLUE AIRCRAFT CAPABILITIES

TYPE ACFT	SPECIFIC MISSIONS		STANDARD FORCE SIZE
A1Ø	INTD CAS		128
F-4	ABA RECCE INTD CAP CAS	AIRDEF DEFSP ESCORT	384
F111	ABA INTD CAS		96

TABLE 2

RED AIRCRAFT CAPABILITIES

TYPE ACFT	SPECIFIC MISSIONS		STANDARD FORCE SIZE
SU7	ABA INTD CAS	DEFSP	112
MIG21	ABA RECCE INTD CAP	CAS AIRDEF	120
MIG23	ABA RECCE INTD CAP	CAS AIRDEF DEFSP ESCORT	320

destroying aircraft on the ground and disrupting base support facilities such as runways, taxiways, or maintenance facilities. All of these actions serve to reduce the enemy's ability to generate sorties. However, several defensive measures are employed to minimize the impact of air base attack. Aircraft shelters and revetments protect aircraft on the ground. Improved surface-to-air missiles (SAMs) and a proliferation of anti-aircraft-artillary (AAA) have posed a serious attrition threat to air base attack aircraft. As a result of these defenses, a complex set of strategies are availabe to the attacker in terms of "mission packages" - protecting the main attack force with escort and/or defense suppression aircraft. The air model in STAG is designed to reflect the impact of mission packages on the outcome of the battle.

The treatment of the air base attack mission in the air model is greatly simplified. Internal computations of the computer are based on highly aggregated interactions between the two opposing forces. Aircraft shelters and revetments are not treated in the model. However, AAA, SAMs, and attrition due to AAA, SAMs and air defenders are modeled. A generalized flow of the attack mission is depicted in Figure 3.

The treatment of attrition from AAA and SAMs embodies a relatively static and predictable array of defenses. AAA loss rates are considered to be the same for all

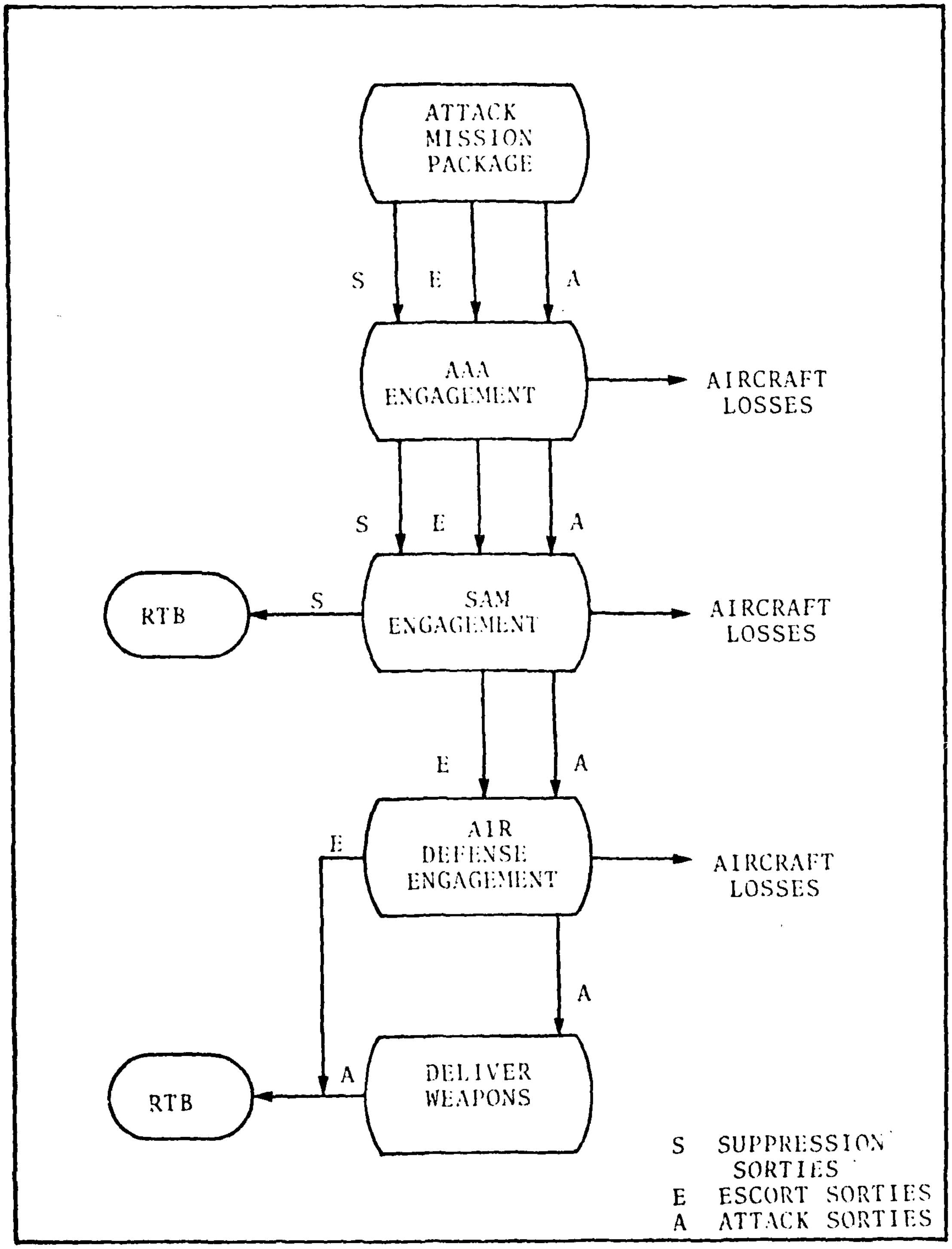

FIGURE 3. MISSION PACKAGE FLOW SEQUENCE

aircraft, and all aircraft must penetrate the coverage of these weapons. SAM units are deployed in two locations: along the FEBA and in an area between the FEBA and the airbases. Suppression aircraft reduce the number of SAM sites available to fire missiles at the attack force.

Attrition due to air defenders is dynamic since the probability of kill depends upon the size of the opposing forces. Escorts serve to reduce the number of available defenders and is discussed in detail below. Surviving airbase attack soties which deliver weapons upon an opponents air field reduce that base's ability to generate sorties.

The maximum number of aircraft which each airbase can support is initially input as data. As the base status is reduced by repeated airbase attack sorties, the number of aircraft which the base can support is proportionately reduced. The amount of reduction is a functions of the number of effective sorties which reach the base and the destructive index for the types of aircraft which attack the base. The users guide (Appendix A) lists the destructive index of each type of aircraft.

Effective sorties are defined as those attackers which survive the AAA and SAM threats and are not detected and engaged by the defenders. Attackers who are detected and engaged by air defense aircraft are assumed to jettison their bomb load. Those attackers detected and engaged have some probability that they will be shot down by the defenders.

INTERDICTION

Interdiction (INTD) missions attempt to damage, destroy, or neutralize support and logistics received by enemy ground units (air bases and army). Destruction of POL and munitions in the logistics pipeline has a more immediate effect on the level of intensity of the conflict than the destruction of command and control facilities or third echelon storage depots.

In the game, interdiction sorties may be split into two components: those that attack air base supply routes and those that attack army supply lines. Interdiction sorties are subject to the same threats experienced by the air base attack sorties. The computer model reacts to successful interdiction sorties flown against an airbase's logistics line by reducing the number of spares that base receives from its sanctuary depot. Each day a base utilization factor is computed by dividing the number of spares on hand by the number of spares required to support the number of aircraft presently located at that base. If the number of sorties that can be supported is less than the number of aircraft located at the base, the player must relocate the excess aircraft or else the model will not permit him to use them on the next day's missions.

In a similar manner, the program reacts to successful interdiction missions against the ground forces by either slowing or accelerating the rate of advance of the FEBA.

Interdiction sorties reduce the number of spares the army receives from its sanctuary depot. A slow down factor is computed each day by dividing the number of spares on hand by the number required to support the current number of divisions engaged in battle.

SAM SUPPRESSION

Defense suppression (DEFSP) missions suppress and destroy enemy ground-to-air defenses in the vicinity of the ground combat zone and the area between the FEBA and the air bases. Aircraft allocated to this mission will reduce ground-to-air losses of other mission aircraft. Employment of suppression aircraft will open a corridor for the attack aircraft to penetrate ground defenses. However, by allocating aircraft to the suppression mission, a commander is using aircraft which might be used for one of the other missions (CAS, ABA, etc.). Hence, a trade-off must be made between reducing losses to the attack force with defense suppression and using those suppression aircraft in a more direct role.

The model views the suppression aircraft as preceeding the main attack force to clear a corridor for these aircraft (see Figure 3). The number of SAM sites encountered by the main attack force is less than the original deployment of SAM sites because of SAM site suppression. This is taken into account by modifying the expected number of SAMs shot at each aircraft by the fraction of SAM sites surviving suppression.

ESCORT

Escort sorties accompany the primary mission aircraft to the target and engage enemy interceptors. Escorts are used as part of a mission package along with defense suppression in an attempt to counter enemy defenses. The cost is in terms of what one must sacrifice to provide the escort package.

Allocating aircraft to the escort missions in STAG will reduce attacker losses due to air defense aircraft. Escort missions can be assigned to accompany the deep penetrators (ABA or INTD) and the interdiction of the army's supply lines. Each escort sortie reduces the effective number of enemy air defense sorties according to a simple subtractive rule. The use of escort sorties is examined more closely in the section on aircraft losses.

AIR DEFENSE

Air defense sorties may be split into two components: those that are deployed forward near the FEBA (CAP) and those that are used for defense of the rear areas (AIRDEF). CAP missions attempt to gain and maintain air superiority by attacking enemy aircraft which enter the forward combat zone surrounding the FEBA. They are used primarily to protect friendly ground forces from enemy CAS sorties and army logistics lines from enemy interdiction. AIRDEF missions are normally on alert; when early warning radar detects an incoming hostile force, the air defenders are "scrambled" to meet the air threat. Air defense also protects friendly air bases and supply lines in the rear of the battle area from enemy air attacks. Air Defense aircraft attack enemy interdiction, air base attack, and reconnaissance aircraft and their escorts. Additionally they reduce the effectiveness of those attackers that survive by causing some aircraft to jettison their munitions.

The effectiveness of an air defense sorties is modeled in STAG using a probability of detection. The likelihood that an air defense aircraft detects an intruder is heavily dependent on the assistance the defensive aircraft receives regarding the location of intruder aircraft. The model attempts to capture the situation in which the air defense search process is essentially autonomous and the probability of detection (P_D) is sensitive to the number of intruders in the friendly air space. Hence, P_D is proportional to the number of opportunities for making a detection. The program also assumes that intruders who are detected and engaged but not shot down will jettison their munitions and return to base.

CLOSE AIR SUPPORT

The army depends on CAS to assist in countering large concentrations of enemy forces. CAS missions attack enemy ground units in actual combat with friendly forces. Air power provides the fastest means of significantly affecting the ground battle. Since most CAS sorties require visual acquisition of ground forces, weather and darkness are significant factors. Normally CAS would be allocated to units faced with a distinct force disadvantage.

Since the model considers only one section of the FEBA, there is no decision on where to allocate the CAS sorties. Weather and darkness are not treated in the model resulting in uniform effectiveness for CAS over the course of the game. The addition of weather or a night cycle would improve the model's treatment of the CAS mission.

RECONNAISSANCE

Accurate intelligence is essential in the successful conduct of an air war especially since resources are limited and attrition is high. Maximum efficiency from limited capability can only be achieved if the information on which decisions are based is timely and accurate.

In STAG information about the status of enemy air bases and ground forces may be obtained through the use of reconnaissance missions. These sorties have no damaging effect on enemy status but are capable of defending themselves if attacked. In order to obtain RECCE information about a particular target, at least one RECCE sortie must survive. For example, if 4 RECCE aircraft were sent against an airbase, and none of them survived, no intelligence information would be delivered on the status of that base. However, if any or all of the 4 returned, the status of the target would be displayed.

REINFORCEMENTS AND LOGISTICS

Beginning on the second day of the game, each side receives the daily aircraft reinforcements listed in Table 3.

BLUE	A-10	SU-7	F-111
	27	51	27
RED	SU-7	MIG-21	MIG-23
	24	24	48

TABLE 3. DAILY AIRCRAFT REINFORCEMENTS

Since each air base starts the game with a surplus of spares and a capability to support more aircraft than it possesses (see Table 4 and 5), the impact of airbase attack and interdiction missions on the overall battle is cumulative and requires several days to manifest itself. Therefore, it would be poor strategy to expend much effort on these missions if it were known that the duration of the game was only going to be a few days.

TABLE 4

BLUE LOGISTICS DATA

AREA	AIRBASE	ARMY
INITIAL FORCE	203 AIRCRAFT	3 DIVISIONS
INITIAL NO. SPARES	250	360
DAILY SPARE UTILIZATION RATE	1.1/A/C	100/ DIVISION
DAILY RESUPPLY RATE (# SPARES)	209	279
MAX SUPPORTABLE A/C PER BASE	220	-

TABLE 5

RED LOGISTICS DATA

AREA	AIRBASE	ARMY
INITIAL FORCE SIZE	187 A/C	9 DIVISIONS
INITIAL NO. SPARES	232	1080
DAILY SPARE UTILIZATION RATE	1.1/A/C	100/ DIVISION
DAILY RESUPPLY RATE (# SPARES)	189	846
MAX SUPPORTABLE A/C PER BASE	220	-

AIRCRAFT LOSSES

Aircraft allocated to the attack of enemy air bases, ground troops, or the logistics network may suffer attrition due to anti-aircraft artillery (AAA) or surface-to-air-missiles (SAMs) located along the FEBA and between the FEBA and the air bases (see Figure 4). The surviving aircraft may then be engaged by enemy air defense aircraft in an air battle where losses are sustained on both sides. Attack aircraft which survive the air defense then proceed to their designated targets.

The ability of the SAM defenses to kill attack aircraft may be reduced by allocating aircraft to the SAM suppression mission. These aircraft precede the main attack force to clear a corridor for the attackers and allow them to penetrate the enemy's defenses. Similarly, allocating aircraft to the escort mission will reduce attack losses due to air defense aircraft. Escort sorties engage the air defenders first and reduce the number of air defenders which can intercept the attackers. SAM suppression and escort missions are critical elements in the air warfare scenario. They serve to reduce overall aircraft losses due to the opponent's defenses.

Air-to-Air losses are computed in terms of the probability of survival of an attack sorties using an exponential of the form:

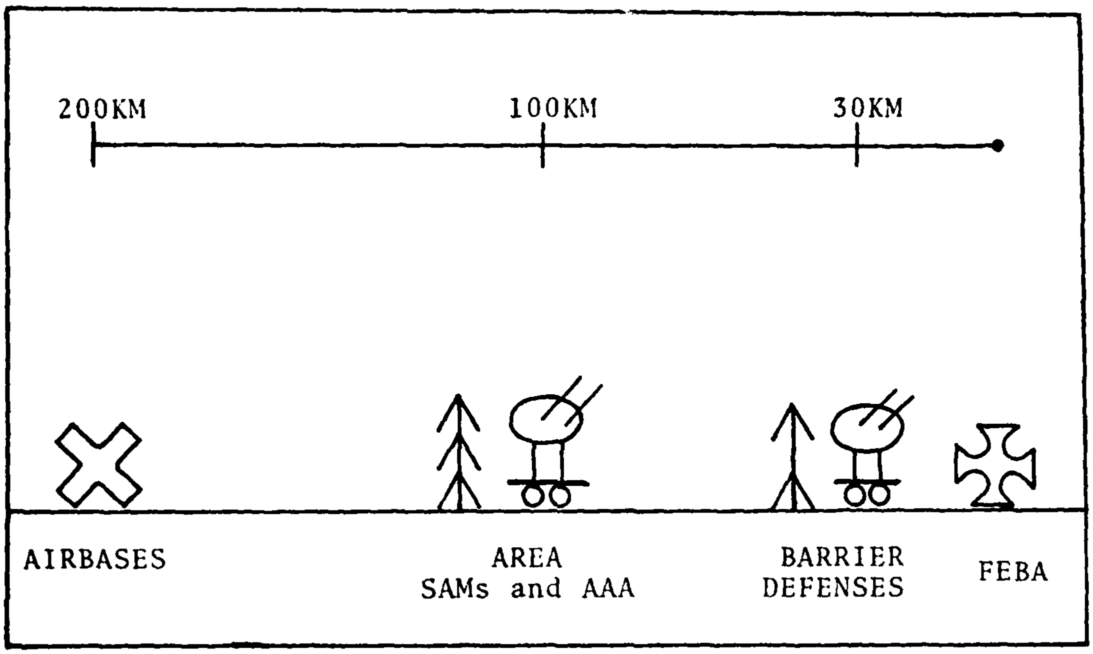

FIGURE 4. GROUND DEFENSES

$$P_{s_A} = e^{-a/b}$$

where,

P_{s_A} = probability an offensive sortie survives

a = function of number of air defense sorties

b = function of number of offensive sorties

a/b = engagement ratio

This exponential form is derived from the Poisson probability distribution and expresses the concept of diminishing returns per weapon because of multiple hits or overlap effects. Hence, the expected number of attackers or defenders killed is not simply proportional to the number of aircraft used. Figure 5 depicts the concept involved. Once the defender has achieved roughly a two-to-one ratio (engagement ratio equals 2) over the attacker, very little is gained in terms of decreasing the attacker's probability of survival. The attacker should attempt to concentrate his forces as much as possible.

One way of achieving this concentration is with the use of escort sorties. The model assumes that each escort sortie will reduce the number of air defenders available to detect and engage a bomber sortie by a specific number according to a simple subtractive rule. Thus if mission A contained 50 bombers and 20 escorts against 30 defenders, only 10 defenders would be eligible to detect and engage the 50 bombers. The other 20 defenders would be occupied by the escorts. Now suppose mission B contained 70 bombers and no escorts against 30 defenders. All of the defenders would be available to detect and engage the bombers. On mission A the engagement ratio of defenders to bombers would be 10/50 (or .2). On mission B the engagement ratio would be 30/70 (or .43). Figure 5 indicates that the probability of survival for bombers on mission A is .82 and the probability of survival for bombers on mission B is .65. Hence, the use of escorts increase the probability of survival of mission A bombers by .17 over mission B bombers.

The air war impacts not only on the air bases and logistics network but also on the ground forces. The next chapter will describe the interactions between opposing ground forces and the impact of air power on the ground battle.

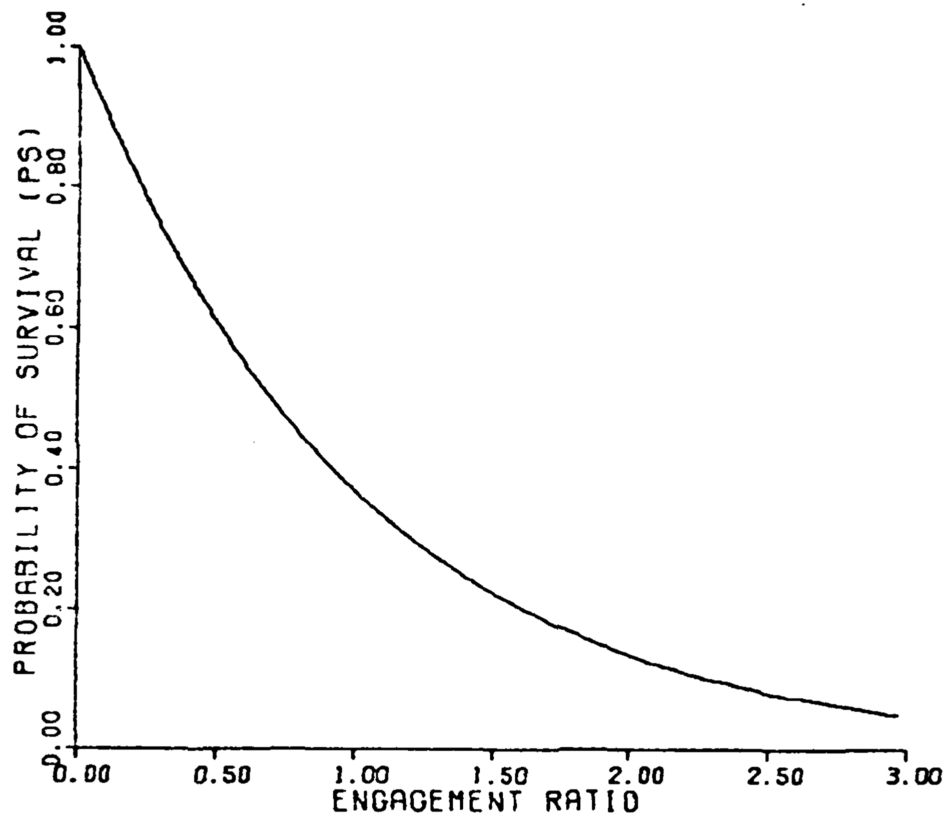

FIG 5 : THE EFFECT OF ENGAGEMENT RATIO ON ATTACKER PS

IV. Ground Warfare

INTRODUCTION

This section describes the submodel used to assess the interaction of opposing ground forces and the effects of combat air support on the ground battle. The model was developed to satisfy two fundamental objectives:

1) to assess the contributions and interactions of air units on the outcome of ground combat, and

2) to employ a relatively simple methodology so that multi-day theater-level wars could be efficiently simulated.

The basic logic of the model is that the *primary purpose* of the red side is to occupy territory while blue is attempting to slow the movement of the forward edge of the battle area (FEBA) as much as possible. The model assumes that there is a fairly well defined FEBA.

COMPOSITION AND CONTROL

The ground force strength available to each player for the conduct of STAG is represented in the form of homogeneous equivalent divisions. Homogeneous refers to the fact that elements of a division, such as artillery, infantry, or tank units, are not explicitly differentiated. Equivalent divisions on both sides means that one blue division has the same firepower or destructive capability as one red division.

The intent of STAG is to consider all front-line ground action as a whole, disregarding individual, localized activity. The initial ground force strength is three divisions for blue and nine divisions for red. There is no ground force augmentation during the game.

The game does not allow players to exercise command decisions in deploying and employing ground forces. As a result, the descriptive detail of the ground operations is not as great as that of the air operations. Concentration of forces for breakthrough purposes is simulated by the initial force sizes. Hence, at the start of the game, it is assumed that the red side has already massed its forces for an attack.

MOVEMENT OF THE FEBA

Since the red army starts the game with a substantial size advantage over the blue army, the FEBA always moves in a forward direction (as viewed by Red). The red side is always advancing, and blue is attempting to slow the movement. The rate of FEBA movement will depend on the relative strengths of the opposing forces. Effective close air support sorties as well as logistics will also influence the rate of FEBA movement. CAS sorties produce casualties in proportion to the destructive index of the aircraft involved (see Appendix A). The index accounts for damage due to disrupting troop coordination, slowing troop movements, and creating an adverse psychological effect on the opponent (Ref 19:30). As mentioned in the previous chapter, a shortage of spares will produce a slowdown of an army's ability to move the FEBA.

A rather simple mathematical expression developed by the Rand Corporation (Ref 8:25) indicates that the average motion of the FEBA may be described using the effective force ratio, F, defined as:

$$F = \frac{M\,(slow) + S_{cas}\,(DI)}{\overline{M}\,(\overline{slow}) + \overline{S}_{cas}\,(\overline{DI})}$$

where

M = the number of attacking division equivalents

$slow$ = the logistics slowdown factor of the attacker

S_{cas} = the number of the attacker's effective CAS sorties

DI = the destructive index of the attacker's CAS aircraft

$\overline{M}$, $\overline{slow}$, $\overline{S}_{cas}$, $\overline{DI}$ = the defender's factors

The daily movement of the FEBA is then expressed as a function of the effective force ratio (see Appendix B for a detailed explanation of this function).

TROOP CASUALTIES

Several possible formulations can be devised to describe troop casualties inflicted by the opposing ground forces. STAG uses a Lanchester-type model adapted from the Rand program, TAGS (Ref 8), in which attrition rates are proportional to the force ratio of the two ground forces. Figure 6 depicts the relationship of offensive and defensive casualties due to ground combat as a function of the force ratio.

The application of air power in the form of close air support will also produce casualties among ground personnel and the destruction of equipment. A simple linear relationship is used to describe the fraction of a division killed by one CAS sortie assuming that the CAS sortie has survived the AAA and SAM threat and has not been shot down by CAP. This linear relationship is a valid treatment assuming that the number of ground targets availabe for attack is large in comparison to the number of CAS sorties flown.

Given this model for the employment of tactical air forces, it was necessary to insure that its behavior was as intended. The next chapter will present an introduction to model verification and validation followed by the specifc procedures used to verify and validate STAG.

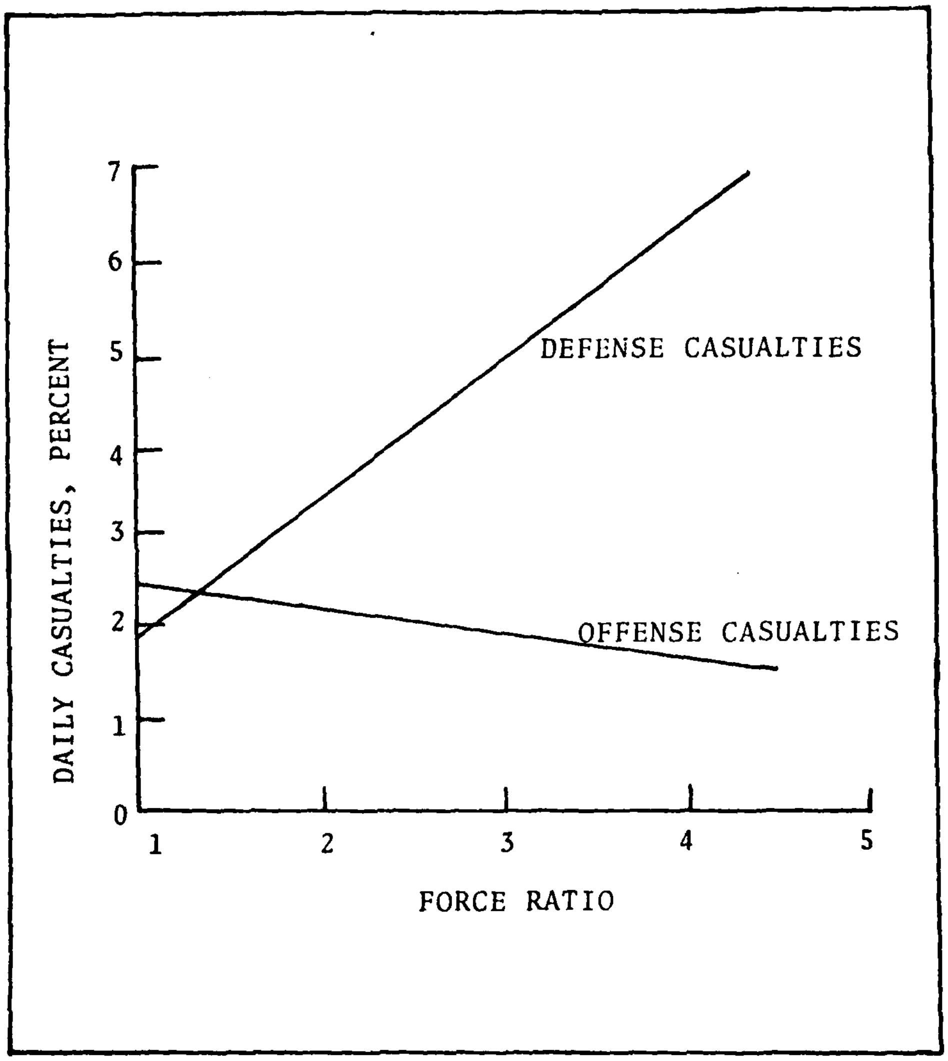

FIGURE 6. RELATIONSHIP OF GROUND CASUALTIES TO FORCE RATIO

V. Verification and Validation

In the development of a simulation model, two of the most important stages the builder must accomplish are verification and validation. Without them the model formulations, preparation, and translation into an acceptable computer language are meaningless. This chapter will present an introduction to the general process of verification and validation followed by the procedures used to verify and validate STAG.

Differentiation between verification and validation is difficult since they are not independent processes. However, verification is generally viewed as insuring that the model behaves the way it was designed. Validation consists of testing the agreement between the behavior of the model and the real system (Ref 18:30). To validate a war game, a means of building confidence in the game's ability to achieve its objectives must be devised. An important distinction between verification and validation is that models can be completely verified, while complete validation is impossilbe. Richard L. Van Horn (Ref 21:247) suggests that a model may be considered valid when it has achieved an acceptable level of confidence. Only the model builder and user can determine what is an acceptable level of confidence.

There are four views concerning the problem of model verification and validation: rationalism, empiricism, pragmatism, and utilitarianism (Ref 18:213). Each of these philosphies will be discussed briefly.

RATIONALISM

Rationalism is closely associated with mathematics and logic. Rationalists contend that a model is simply a system of logical deductions derived from a set of unquestionable truths. Immanual Kant used the term "synthetic a priori" to describe these premises of unquestionable truth (Ref 13: B-93). Kant and his followers argued that if one accepts the basic premises about a model (which they considered unquestionable) and the formal logic used to deduce the consequences, then one accepts the validity of the model. The problem of verification has then been reduced to the problem of stating the basic assumptions underlying the behavior of the system being modeled.

Unfortunately many of the "synthetic a priori" premises proposed by rationalists are not at all obvious. The premises themselves often reveal the questionable nature of their obviousness. Consider, for example, the premise that more money spent by government will result in increased public service. This premise ignores the impact of increased governmental spending on inflation or the misapproriation of government funds due to fraud. Countering the beliefs of the rationalists, empiricists deny the very existence of "synthetic a priori" premises.

EMPIRICISM

In direct contract to rationalism, empiricism refuses to accept any assumption that cannot be verified by experiment or analysis of statistical data (Ref 18:214). Empiricists insist that model verification must begin with facts not assumptions. Hence, they regard empirical science, and not mathematics, as the ideal form of knowledge. "A sentence the truth of which cannot be determined from possible observation is meaningless" (Ref 17:256). Empiricists often employ formal statistical tests of hypothesis, based on historical data, to validate a model. Rationalists argue that historical data often does not apply to dynamic systems and that statistical tests do not show that a hypothesis can be accepted, only whether or not it can be rejected. The controversy is over matters of emphasis - what a model should be founded upon. A less extreme point-of-view is held by the third group, the pragmatists.

PRAGMATISM

While both the rationalist and the empiricist are primarily concerned with the internal structure of the model, disagreeing over the nature of the internal relationships that are valid, the pragmatist feels that the validity of a model depends upon its ability to properly transform inputs into outputs. If the model fulfills the purpose for which it was built, then it is a valid model. Proposing that the usefulness of the model be the key to its validation, pragmatists emphasize the question of whether errors in the model render it too weak to serve its intended purpose.

UTILITARIANISM

Perhaps the most practical approach to model verification and validation is taken by the utilitarian. Two important characteristics of this approach are:

1. The objective is to validate a specific set of insights not necessarily the mechanism that generated the insights.
2. There is no such thing as "the" appropriate validation procedure. Validation is problem-dependent. (Ref 21:248)

Hence, this approach advocates the use of any of the verification and validation tests which might apply to the model being tested. The following section describes some of the specific tests which were used to verify and validate STAG.

VERIFICATION TESTS

The following verification tests were used to demonstrate that there are no logical or calculational errors in the computer program (Ref 9:119). The first test of the model was to demand that its behavior not be obviously implausible. In the early development of the model, the implausible results are apt to be of a gross nature. For example, in a tactical situation the model may indicate that more aircraft returned from a mission than originally started. In a sense the program could be "creating" aircraft. These errors were quite easy to detect and rectify.

Another effective test is to attempt to force additional obvious inadequacies with extreme levels or flows in the system. Model behavior is more unpredictable under normal operating conditions. For example, in STAG the results were observed when no aircraft on either side were launched. The results from such a strategy were easy to predict, since there should have been no losses or damage on either side; this was the actual result.

Once the obvious errors were eliminated, attention was directed at more subtle performance. The multiple mode test considered whether or not the model would provide different behavior when presented with different inputs. In applying this test specifically to STAG, the following experiment was employed. Three airbase attack scenarios were designed to demonstrate that decreasing the number of air defenders at a base in each scenario would result in a corresponding increase in the destruction of the airbase and a decrease in the number of attack aircraft lost. In the first scenario blue had fifty aircraft defending its airbase; red had fifty aircraft attacking this base (Table 6). In the second scenario blue had forty aircraft defending its airbase; the red attack force remained unchanged. Table 6 indicates that red lost one less aircraft and increased his destruction of the blue airbase. In the third scenario, blue had thirty defending aircraft; again the red attack force remained unchanged. Results of this experiment confirmed that different inputs do indeed produce different behavior.

SCENARIO	#Blue DEFENDERS	#RED ATTACKERS	#RED A/C LOST	BLUE BASE STATUS
1	50	50	8	.958
2	40	50	7	.949
3	30	50	6	.938

TABLE 6. Results of Multiple Model Experiment

VALIDATION TEST

The objective of STAG is to demonstrate the principles of tactical air warfare to the players. Specifically, a desired result of the model is that the use of escort sorties and SAM suppression sorties will result in a lower aircraft loss rate and a higher number of bombers reaching the target. To test whether or not the model correctly demonstrated this principle, a two-factor, two-level, full factorial experiment was used. In this experiment the two factors were: the use of escort sorties and the use of suppression sorties. There were four possible combinations of these factors:

(a) no escort/no suppression

(b) no escort/ with suppression

(c) with escort/ no suppression

(d) with escort/with suppression

The results are depicted in Table 7.

SCENARIO	AIRCRAFT LOSS RATE	NUMBER OF EFFECTIVE SORTIES
No Escort/No Supp	.34	28
No Escort/With Supp	.25	36
With Escort/No Supp	.28	38
With Escort/With Supp	.15	52

TABLE 7. Results of Validation Experiment

The number of aircraft involved in each scenario was 150 bombers, 84 escorts, 84 SAM suppressors, and 150 air defenders. Effective sorties were computed with the following equation:

$$\left(\begin{array}{c}\text{NUMBER}\\\text{EFFECTIVE}\\\text{SORTIES}\end{array}\right) = \left(\begin{array}{c}\text{NUMBER}\\\text{BOMBERS}\end{array}\right) - \left(\begin{array}{c}\text{NUMBER BOMBERS}\\\text{KILLED BY GROUND}\\\text{TO-AIR WEAPONS}\end{array}\right) - \left(\begin{array}{c}\text{NUMBER BOMBERS}\\\text{DETECTED \& ENGAGED}\\\text{BY DEFENDERS}\end{array}\right)$$

The assumption is made here that if a bomber is detected and engaged by air defenders, it will jettison its bombs and hence will become ineffective.

The results of this experiment clearly indicate that the model exhibits the desired relations among attackers, escorts, and suppressors. The experimental results do increase confidence in the model and hence serve to validate it. Since STAG is an interactive game, most people will agree to placing a higher "a priori" confidence on a man than on a model of him. In other words, the game would probably hold less face validity if it were played against a pre-programmed strategy instead of another person.

The most sensible approach to model verification and validation is the utilitarian approach. This methodology recommends the use of any verification and validation technique which seems appropriate to the problem. As difficult as verification and validation are for models in general, they are particularly nebulous for tactical air war games. Very little literature exists on the subject, and the modeler is required to devise appropriate techniques to handle a particular situation. Validation is a never ending process.

VI. Summary, Conclusions, and Recommendations

STAG and the accompanying documentation represent the accomplishment of the objectives of this research. The model is a highly aggregated game comprised of interaction equations which utilize the allocation of aircraft to various missions to obtain the outcome of an offensive versus defensive systems engagement. The simulation which supports the model consists of an interactive air portion and a parametric ground portion. The theater of operations consists of two sides with their respective air and ground forces.

The model contains several features unique among war games. The ability to input the length of the game gives players increased flexibility in formulating new strategies. A player must be able to adjust his strategy to accomodate either a long or short war. Although it is a game, STAG has definite worth as an educational tool.

The model developed in this study was designed to provide individuals an opportunity to plan and conduct an air war and to test various air employment concepts. Existing war games are quite large and contain so many factors that the main effects of a player's employment decisions are confounded by the interactive effects of the factors. The war game described herein, STAG, has a limited number of factors so that it is easier for players to determine the main effects of their strategies. Players can trade-off numbers and types of aircraft on various missions and observe the impact of these decisions on the outcome of the battle.

The idea of including details of only those factors that are of immediate interest in a particular model suggests the development of a family of models, each appropriate for a specified level of detail. Development of such a heirarchy of war games is an area for future reserach. When STAG has been exposed to a wide audience many areas for improvement will be found. Some of these areas are listed below:

(1) inclusion of weather, a night cycle, and precision munitions

(2) a variable game length which is stochastically determined

(3) an alternative to an end game score to discourage players from "gaming the game"

(4) inclusion of survivability indices for the various aircraft

(5) a more effective way of handling player force inputs

(6) sensitivity analysis on model parameters

Using STAG as a median, one could develop an even simpler game with one airbase on each side for example. Another possibility which expands on the detail presented in STAG is the inclusion of a number of different factors (listed above in item one).

A variable game length would be much more realistic. Instead of allowing the player to determine the length of the game, the program would stochastically determine the game length. One day prior to end of the game the program would inform both players that the game will terminate in one day. This modification would discourage extreme strategies employed when the duration of the game is known from the start.

One suggestion for an alternative to an end game score would be the use of a stopping rule devised by the analyst. The game would progress until one side has met or exceeded the criterion set up in the program.

The survivability of each type of aircraft is the same in the present version of STAG. The use of survivability indices to reflect a particular aircraft's ability to survive in a hostile environment would be a more realistic treatment of the situation.

The number of aircraft a player may input at the beginning of the game is currently limited by the data structure. Associated with each base is a maximum number of aircraft which that base can support. Changing this value in the data structure to a percentage of the initial force size would add flexibility to the model.

Performing sensitivity analysis on the model's parameters would be intellectually appealing. It might be interesting to attempt to determine which factors in the model have the most significant impact on the outcome of the game. Readers who are interested in this area should consider the use of multivariate analysis as deomstrated by Danial Nussbaum (Ref 15).

STAG has the potential for being the basis for the development of a family of war games to teach the principles of tactical air warfare. The possibilities and opportunities in this area are almost limitless.

APPENDIX A

User's Guide for STAG

APPENDIX A

User's Guide for STAG

This section is designed to acquaint players with the rules, procedures, and pecularities of the game. STAG was designed to be played interactively and was implemented in FORTRAN IV on the CDC Cyber 175 INTERCOM system. The reader should consult the annex to Appendix A while reading this guide for a sample listing of data input.

1. Initially the program will request an input for the number of days the game will last. The suggested minimum and maximum length is two and five days respectively; but the program will accept any number of days as an input. Being able to select the length of the game provides a unique feature not found in other war games. A player's strategy must now be partially based on the fact that he knows when the war game will end, and he knows how many days are left in the war.

2. At the end of the game, the program computes a player's score based on the weights displayed at the beginning of the game (see annex). To accomodate players whose utility is different from the programmed weights, players have the option of changing the emphasis placed on FEBA movement and aircraft exchange ratio. For a more detailed description of the scoring system consult Appendix B.

3. The next option offered by the program is the selection of aircraft force size. If a player selects the standard force input (by entering "1"), the aircraft beddown listed in the annex will be displayed. Players desiring to create their own forces can do so by entering "2", and the program will prompt for inputs of force size by type of aircraft and base for each side. If either player chooses to input his own force, then both players will be required to input their own force list.

4. After selecting aircraft force size, the program will list aircraft beddown for each player.

5. To provide a means of preventing an opponent from viewing "privileged" information, the program will print the message, "Enter 1 to continue" and stop after each side has completed an activity. That player should remove his information by tearing off the output. The other player should then enter "1" (actually any value will be accepted) to complete his turn. This procedure is especially critical for sortie allocation since an unfair advantage is gained by seeing an opponent's strategy.

6. Every time aircraft beddown is displayed, players will have the option of moving aircraft from one base to another. When both players are satisfied with the location of their aircraft, sortie allocation will begin.

7. Sortie allocation is the most important aspect of the game. The example in the annex will be explained in detail. Players should consult Tables 1 and 2 (Chapter III) to determine which missions their aircraft are capable of performing. Sorties are allocated by type aircraft, mission, and target. The term "TGT" beneath the target column requests the total number of that type aircraft a player wants to allocate to that mission.

8. In the example, blue has allocated a total of 80 A-10 aircraft (from his available force of 128) to interdiction. The second line then asks how many of those 80 aircraft blue wants to send against red base one (RB1). The example shows that blue has sent 20 A-10 aircraft against each target (RB1, RB2, RB3, RARMY). Since there are no designated targets for the close air support (CAS) mission (also the case for the combat air patrol (CAP) mission), blue only allocated a total number of aircraft to CAS. The program will not allow a player to allocate more aircraft than the number available listed under the column MAX AVAILABLE. However, it will not prevent a player from losing sorties by allocating too few aircraft.

9. Air defense and defense suppression missions require additional comments. Although the program lists the opponents' bases under the TARGET column for the air defense mission, these sorties will be used to protect the friendly bases

from attack. In the example blue has allocated 40 F-4 sorties for the air defense of each "target" (RB1, RB2, RB3). Actually these sorties will be used for the air defense of BB1, BB2, and BB3. Defense suppresssion sorties allocated to targets RB1, RB2, and RB3 in the example are used for suppression of area SAMs. Sorties allocated to the target RARMY are used for suppression of FEBA barrier SAMs.

10. After a player has finished allocating his sorties, the program will list the number of sorties allocated to each mission and the percentage of effort allocated to each mission.

To assist players in deciding which aircraft are most effective on a particular mission, tables A-I and A-II list the destructive indices of aircraft capable of performing the CAS and air base attack (ABA) mission. Interdiction results depend on the number of aircraft attacking a target and not the type of aircraft. Hence, there are no destructive indices listed for this mission.

TABLE A-I

CAS DESTRUCTIVE INDICES

AIRCRAFT TYPE	DESTRUCTIVE INDEX
A-10	.0003
F-4	.00015
F-111	.0003
SU-7	.00015
MIG-21	.0001
MIG-23	.00015

TABLE A-II

ABA DESTRUCTIVE INDICES

AIRCRAFT TYPE	DESTRUCTIVE INDEX
SU-7	.0015
MIG-21	.001
MIG-23	.002
F-4	.002
F-111	.003

```
    SEED=  14.08.07.
(1)--->   ENTER NUMBER OF DAYS THAT YOU WISH TO PLAY(MINIMUM OF 2, MAXIMUM OF
    5)....
       45000B   CM STORAGE USED
          4.931 CP SECONDS COMPILATION TIME1
          PLAYERS SCORES ARE BASED ON THE FOLLOWING WEIGHTS:
(2)------------------>   FEBA MOVEMENT...50%
                         A/C LOSS RATIO..50%
          DO YOU WANT DIFFERENT WEIGHT FACTORS?? TYPE ...1FOR YES...TYPE....0
    FOR NO3
(3)--->   ENTER 1 FOR STANDARD FORCE INPUT
          ENTER 2 FOR FORCE INPUT BY PLAYER1
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
(4)------------------>    BLUE  AIRCRAFT BEDDOWN
    ---------------------------------------------
          BASE            TYPE            TOTAL
                                          ACFT
    ---------------------------------------------
          BB1             A10               48
          BB1             F4               128
          BB1             F111              32
    ---------------------------------------------
          BB2             A10               32
          BB2             F4               128
          BB2             F111              32
    ---------------------------------------------
          BB3             A10               48
          BB3             F4               128
          BB3             F111              32
    ---------------------------------------------
    ---------------------------------------------
              TOTAL:                       608
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

(5)-> ENTER 1 TO CONTINUE
```

```
1
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
               RED   AIRCRAFT BEDDOWN
----------------------------------------------
    BASE          TYPE          TOTAL
                                ACFT
----------------------------------------------
    RB1           SU7             32
    RB1           MIG21           40
    RB1           MIG23          104
----------------------------------------------
    RB2           SU7             43
    RB2           MIG21           40
    RB2           MIG23          104
----------------------------------------------
    RB3           SU7             32
    RB3           MIG21           40
    RB3           MIG23          120
----------------------------------------------
----------------------------------------------
         TOTAL:                  580
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

ENTER 1 TO CONTINUE

1
DOES BLUE  WISH TO MOVE A/C FROM ONE BASETO ANOTHER??  TYPE 1 FOR YES...
TYPE 0 FOR NO....0
DOES RED   WISH TO MOVE A/C FROM ONE BASETO ANOTHER??  TYPE 1 FOR YES...
TYPE 0 FOR NO....0
```

(6)

⑦ → BLUE SORTIES WILL BE ALLOCATED NOW
ENTER NO. OF BLUE SORTIES BY TYPE AIRCRAFT TO EACH MISSION(AS IT IS DISPLAYED)

AIRCRAFT ALLOCATED TO EACH BASE (UNDER HEADING OF...TARGET) FOR AIR DEFENSE WILL BE USED TO PROTECT THAT AIRBASE FROM ATTACK.
FOR EXAMPLE,IF YOU ONLY HAVE 20 F4 A/C AT BASE 2 YOU CANNOT ALLOCATE MORE THAN 20 F4 A/C TO THE AIR DEFENSE OF BASE 2.

SIDE	TYPE A/C	MISSION	MAX AVAIL	TARGET	
BLUE	A10	INTD	128	TOT	....80 ← ⑧
BLUE	A10	INTD	80	RB1	20
BLUE	A10	INTD	60	RB2	20
BLUE	A10	INTD	40	RB3	20
BLUE	A10	INTD	20	RARMY	20
BLUE	A10	CAS	43	TOT	43
BLUE	F4	ABA	384	TOT	90
BLUE	F4	ABA	90	RB1	30
BLUE	F4	ABA	60	RB2	30
BLUE	F4	ABA	30	RB3	30
BLUE	F4	RECCE	294	TOT	24
BLUE	F4	RECCE	24	RB1	8
BLUE	F4	RECCE	16	RB2	8
BLUE	F4	RECCE	8	RB3	8
BLUE	F4	INTD	270	TOT	8
BLUE	F4	INTD	8	RB1	8
BLUE	F4	CAP	222	TOT	40
BLUE	F4	CAS	222	TOT	0
BLUE	F4	AIRDEF	222	TOT	120
BLUE	F4	AIRDEF	120	RB1	....40 ← ⑨
BLUE	F4	AIRDEF	80	RB2	40
BLUE	F4	AIRDEF	40	RB3	40
BLUE	F4	DEFSP	102	TOT	64

```
BLUE     F4      DEFSP     64      RB1  ....16
BLUE     F4      DEFSP     48      RB2  ....16
BLUE     F4      DEFSP     32      RB3  ....16
BLUE     F4      DEFSP     16      RARMY....16
BLUE     F4      ESCORT    33      TOT  ....33
BLUE     F4      ESCORT    33      RB1  ....9
BLUE     F4      ESCORT    29      RB2  ....9
BLUE     F4      ESCORT    20      RB3  ....9
BLUE     F4      ESCORT    11      RARMY....11
BLUE     F111    ADA       96      TOT  ....96
BLUE     F111    ADA       96      RB1  ....32
BLUE     F111    ADA       64      RB2  ....32
BLUE     F111    ADA       32      RB3  ....32
MISSN:   ADA   RECCE   INTD   CAP   CAS   AIRDEF  DEFSP   ESCORT   TOT
(10)-->  186     24     88     40    48    120     64      38      608
PERCENT .306   .039   .145   .066  .079   .197   .105    .063    1.000

ENTER 1 TO CONTINUE
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

APPENDIX B

Analytical Formulations

APPENDIX B

Analytical Formulations

AIR EQUATIONS

The interaction equations for the offensive/defensive engagements in the air battle are presented below in the approximate order of their occurence in the program. Many of the approaches used in the air model were adopted from routines in the Lulejian theater-level model (Ref 14), while the ground model structure was adopted from the Rand model TAGS (Ref 8).

1. SAM Suppression and AAA

The following assumptions apply:

(a) Suppression aircraft precede attack aircraft.

(b) Sites being suppressed get first shot at attackers.

(c) SAM sites are suppressed for one day only. A specific fraction of those sites suppressed are assumed destroyed.

Equations for SAM Suppression mission:

$$\begin{bmatrix}\text{Expected Number}\\\text{of SAMs shot at}\\\text{Suppressor Aircraft}\end{bmatrix} = \begin{bmatrix}\text{Fraction of FEBA}\\\text{Barrier Covered}\\\text{by SAM Sites}\end{bmatrix} \times \begin{bmatrix}\text{Probability a SAM}\\\text{Site acquires a}\\\text{Penetrator}\end{bmatrix} \times \begin{bmatrix}\text{Simultaneous}\\\text{Missile Firing}\\\text{Capability of}\\\text{the SAM Site}\end{bmatrix} \times 1/4$$

where

$$\text{Fraction of FEBA Barrier Covered by SAM Sites} = \left[1 - \exp\frac{-(\text{number of Barrier SAM Sites})(2)(\text{SAM Site Radius})}{(\text{Length of Barrier SAM Deployment})}\right]$$

$$\text{Number of Aircraft Surviving SAM Site Suppression Mission} = \text{Number of Aircraft that Survive to Perform SAM Suppression} \times \left[1 - \text{Single shot Prob a SAM Kills a Supp}\right]^{(\text{Expected No. of SAM Shot})}$$

$$\text{Number of Aircraft that Survive to perform SAM Suppression} = \left[\text{Number of A/C Allocated to Suppression}\right] \times \left[\text{Probability of Surviving Deployed AAA}\right]$$

$$\text{Number fo SAM Sites Suppressed by Suppresion A.C} = \left[\text{Number of SAM Sites}\right] \times \left[1 - \exp\frac{-(\text{number of suppressors})}{(\text{number of SAM Sites})}\right]$$

$$\text{Number fo SAM Sites Destroyed by Supression A/C} = .3 \times \left[\text{Number of SAM Sites Suppressed}\right]$$

$$\text{Number of Attack Aircraft Surviving SAM Fire} = \left[\text{Number of Attack A/C Entering SAM Area}\right] \times \left[1 - \text{Single Shot Prob a SAM Kills a Attacker}\right]^{(\text{Expected No. of SAM Shot})}$$

where

$$\text{Expected No. of SAMs Shot} = \left[\text{Expected No. of SAMs Shot at Supp A/C}\right] \times \left[1 - \text{Fraction of SAM Sites Suppressed}\right]$$

2. Air Defense

$$\text{Defending A/C Engagement Potential} = \left[\text{Number of Defending A/C}\right] \times \left[\text{Potential of a Defending A/C to Detect and Engage an Offensive A/C}\right]$$

$$\text{Number of Attackers Detected and Engaged} = \left[\text{Number of Attackers Surviving Ground Defenses}\right] \times \left[\text{Probability an Attacker is Detected and Engaged by a Defender}\right]$$

$$\text{Probability and Attacker is Detected and Engaged} = \left[1 - \exp\frac{-(\text{Defending A/C Engagement Potential})}{(\text{Number of Attackers Surviving})}\right]$$

$$\text{Prob an attacker is killed by a Defender} = \left[1 - \exp\frac{-(\text{Defender } P_k)(\text{No. of Defenders Engaged})}{(\text{Number of Attackers Engaged})}\right]$$

$$\text{Prob a Defender is killed by an Attacker} = \left[1 - \exp\frac{-(\text{Attacker } P_k)(\text{No. of Attackers Engaged})}{(\text{Number of Defenders Engaged})}\right]$$

3. Aircraft Losses

The program uses the subroutine RANDOM to determine the number of aircraft killed given the number of aircraft engaged and the probability an aircraft is killed. RANDOM computes kills using a binomial criterion. Each encounter is treated as an independent Bernoulli trial. For each encounter a random number is drawn; if the random number is less than the P_k of the attacker, the aircraft is considered killed. Otherwise the aircraft survives but is assumed to have jettison its ordinance load. The attacking aircraft which survive the ground-to-air defenses and are not engaged by the air defenders are sent against the opposing air base or logistics system for the final computations. For the air base attack mission, base status, STAT, is reduced in the following manner:

$$\text{STAT} = \text{STAT} - \sum_{i=1}^{3} (f_i n_i)$$

where,

STAT = base status

f_i = destructive index for type i aircraft

n_i = number of type i aircraft

The number of sorties an airbase can support is then computed:

NSORT = MAX X STAT

where,

NSORT = number of sorties an airbase can support

MAX = maximum number of sorties an airbase can can support if fully operational

STAT = base status

Interdiction sorties reduce the number of spares received by an opponent in the following manner:

$$SPARES_i = SUPPLY \times (.995)^{INTD_i}$$

where,

SPARES = number of spares received daily by base i

SUPPLY = maximum supply capability of the logistics network

INTD = number of effective interdiction sorties against base i

i = 1 for Base 1

i = 2 for Base 2

i = 3 for Base 3

i = 4 for Army

GROUND EQUATIONS

The daily movement of the FEBA, FVEL, is expressed as a function of the effective force ratio, F, in the form:

$$FVEL = VMAX \left\{ SIN\left[\frac{\pi}{2}\left(\frac{F - X_1}{X_2 - X_1}\right)\right] \right\}^{2X_3}$$

where,

VMAX = maximum velocity of the FEBA against negligible opposition X_1, X_2, X_3 are constants input by the analyst. (Ref 8: 11

Figure B-1 indicates how the movement rate is affected by selection of the constants. In the present version of STAG, the value of the constants X_1, X_2, and X_3 have been adapted from the Rand model TAGS to achieve approximately the same FEBA movement in comparable situations as achieved by more detailed ground warfare models.

Daily troop casualties inflicted by CAS are a function of the number and type of aircraft involved. The total casualties per day produces by CAS sorties, C_{cas}, is given by

$$C_{cas} = \bar{M} \left[1 - \exp \sum_{i=1}^{3} \left[D_i S_i / \bar{M} \right] \right]$$

where

$\bar{M}$ = number of enemy divisions

D_i = destructive index for type i aircraft

S_i = number of successful friendly CAS sorties of type i aircraft

SCORING SYSTEM

The score given to each player at the end of the game is computed using two results of the game: the cumulative FEBA movement and the aircraft exchange ratio for each side Unless the players choose differently, the two factors are weighted equally in computing a total score:

FEBA Movement weight is 50 percent

Exchange Ratio weight is 50 percent

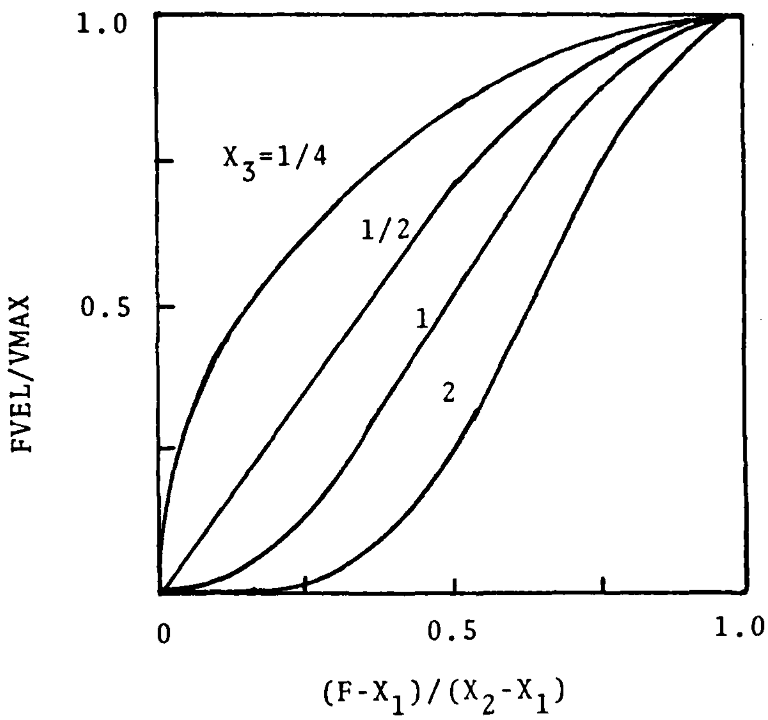

FIGURE B-1. Effect of Selected Constants on FEBA Movement Rate

The program fisrt computes a FEBA ratio, FRATIO, based on the ratio of cumulative FEBA movement to nominal FEBA movement:

$$FRATIO = \frac{TFEBA}{NOM}$$

where,

TFEBA = the cumulative FEBA movement during the game

NOM = the nominal FEBA movement listed in the program

To determine the values of the nominal FEBA movement contained in the program, the following analysis was conducted. First, the model was run with no CAS sorties allocated by either side. Table B-2 depicts the cumulative FEBA movements over a five day period under three different scenarios. The last column lists the nominal values computed using the other three.

DAY	NO CAS	BLUE CAS	RED CAS	NOMINAL
1	7.31	7.16	7.34	7.25
2	15.02	14.66	15.19	14.92
3	23.23	22.54	23.62	23.08
4	31.96	30.83	32.72	31.76
5	41.29	39.59	42.42	40.98

FIGURE B-2 CUMULATIVE FEBA MOVEMENT UNDER CONTROLLED SCENARIOS

The values listed under the blue and red CAS colum were produced by allocating 25 percent of that side's force to the CAS mission each day and noting FEBA movement. The nominal values were computed from the following equations:

$$\frac{NOM}{BMOV} = \frac{RMOV}{NOM} \quad \text{or,} \quad NOM = \sqrt{(BMOV)(RMOV)}$$

where,

NOM = the nominal FEBA movement

BMOV = FEBA movement with no red CAS and a set percentage (25%) of blue's forces allocated to CAS

RMOV = FEBA movement with no blue CAS and a set percentage (25%) of red's forces allocated to CAS

The values listed in the first column (no CAS) could not be used to compute scores for FEBA movement because the structure of the model makes it easier for blue to slow the movement of the FEBA than for red to accelerate it. Scores based on values would be based in favor of the blue side. By using nominal values as a standard of comparison, the score for FEBA movement reflects the degree to which the ground support portion of one player's strategy was superior/inferior to his opponent's.

Once the FEBA ratio has been computed, it is a simple matter to compute each player's score for the ground portion of the game:

RED GROUND SCORE = FRATIO X 50

BLUE GROUND SCORE = 1/FRATIO X 50

Scores for the air portion of the game are computed using the loss ratios for each side:

$$BRATIO = \frac{BLOST}{BTOT}$$

$$RRATIO = \frac{RLOST}{RTOT}$$

where,

BRATIO, RRATIO = loss ratio for each side

BLOST, RLOST = number of aircraft lost by each side

BTOT, RTOT = original number of aircraft plus daily reinforcements for each side

The program then computes the Echange Ratio, ERATIO, defined as the blue loss rate divided by the red loss rate:

$$ERATIO = \frac{BRATIO}{RRATIO}$$

Exchange ratio has traditionally been used to express relative success in air-to-air combat in terms of enemy aircraft killed per friendly aircraft killed. The exchange ratio used in STAG is a slight variation on the traditional interpretation. Scores for the air portion are computed as follows:

RED AIR SCORE = ERATIO X 50

BLUE AIR SCORE = 1/ERATIO X 50

Total scores are then computed by summing the air and ground scores.

It should be noted that the measures of merit used in this game are but two of many possibilities. Depending upon the situation and the utility of the commander, a host of other measures would be equally valid. Examples include rate of kill, force drawdown, and enemy casualties.

Appendix C

Source Listing of Program

```
C     ***         GLOSSARY          ***
C
C
C     *** THE FOLLOWING IS A LIST OF THE PROGRAM'S ***
C     *** MAJOR ARRAYS AND VARIABLES AND THEIR ***
C     *** MEANINGS. NAMES FOLLOWED BY A  $  HAVE ***
C     *** A DUAL COUNTER PART FOR THE RED SIDE. FOR ***
C     *** EXAMPLE, THE COUNTER PART FOR THE ARRAY BARMY ***
C     *** IS THE ARRAY RARMY ***
C
C
C
C     *** AV[illegible].....NOMINAL CUMULATIVE DAILY FEBA ***
C     ***            MOVEMENT                    ***
C     *** BA$.........NO. BLUE DIVISIONS REMAINING  ***
C     ***            AFTER CURRENT DAY'S LOSSES    ***
C     *** BARMY(I)$...BLUE ARMY ARRAY:              ***
C     ***            1=ORIGINAL NO. DIVISIONS      ***
C     ***            2=CURRENT NO. DIVISIONS       ***
C     ***            3=NO. SPARES ON HAND          ***
C     *** BBASE(K)$...NAME OF BLUE FORCES: TOT,BB1, ***
C     ***            BB2,BB3,BARMY                 ***
C     *** BFORCE(I,M,K)$.. BLUE FORCE ARRAY-- NO. OF***
C     ***            TYPE I A/C FLYING MISSION M   ***
C     ***            AGAINST TARGET K              ***
C     *** BFRAC(I)$...DESTRUCTIVE INDEX ARRAY FOR   ***
C     ***            TYPE I A/C ON CAS             ***
C     *** BITORY(K)$..NO. SPARES AT EACH AIRBASE I  ***
C     *** BLOSS(I,M,K)$..BLUE LOSS ARRAY--NO. OF    ***
C     ***            TYPE I A/C LOST FLYING MISSION***
C     ***            M AGAINST TARGET K            ***
C     *** BLOST(I)$..NO. BLUE DIVISIONS LOST TODAY  ***
C     *** BSAM(I,K)$.BLUE SAM ARRAY FOR TYPE I SAM  ***
C     ***           WITH CHARACTERISTIC J          ***
C     *** BSLOW$.....SLOWDOWN RATE FOR BLUE ARMY    ***
C     *** BSTAT(I)$.STATUS OF BLUE BASE I          ***
C     *** BTYPE(I)$.BLUE AIRCRAFT TYPE I           ***
C     *** CRATIO....CURRENT FORCE RATIO            ***
C     *** DAY........CURRENT DAY OF THE WAR      ***
C     *** ENMS.......EXPECTED NO. AREA SAMS SHOT  ***
C     *** ENSS.......EXPECTED NO. FEBA SAMS SHOT  ***
C
C     *** FACT.......ABA DESTRUCTIVE INDEX FOR RED ***
C     ***            A/C I=1,2,3--FOR BLUE A/C I=3,4 ***
C     *** FRATIO.....FORCE RATIO AT BEGINNING OF DAY ***
C     *** IBFORCE(I,M,K)$..STANDARD INPUT BLUE ARRAY ***
C     *** IDAY.......NO. DAYS THE WAR WILL LAST ***
C     *** IWT(I)........WEIGHT OF FEBA MOVEMENT AND A/C ***
C     ***            LOSS RATIO FOR SCORING SYSTEM   ***
```

```
C     *** MISSN......NAME OF MISSIONS ***
C     *** MOST.......MAXIMUM RESUPPLY RATE ARRAY I=1 ***
C     ***            FOR BLUE I=2 FOR RED; J=1 FOR   ***
C     ***            AIRBASES J=2 FOR ARMY           ***
C     *** NADD.......NUMBER A/C ADDED TO INVENTORY   ***
C     ***            EACH DAY                        ***
C     *** NAS(I)........NUMBER OF ATTACK SORTIES        ***
C     *** NBACLS$....CUMULATIVE TOTAL OF A/C LOSSES  ***
C     *** NBIEF$......NUMBER OF EFFECTIVE BLUE INTD   ***
C     ***            SORTIES AGAINST ARMY            ***
C     *** NBIF(I)$..NUMBER OF EFFECTIVE BLUE INTD    ***
C     ***           SORTIES AGAINST RED BASE (I)     ***
C     *** NBREF$.....NUMBER OF EFFECTIVE BLUE RECCE   ***
C     ***           SORTIES AGAINST ARMY             ***
C     *** NBRF(I)$..NUMBER OF EFFECTIVE BLUE RECCE   ***
C     ***           SORTIES AGAINST RED BASE (I)     ***
C     *** NBSORT(I)$..MAX NO. BLUE SORTIES AVAILABLE ***
C     ***           FROM BASE (I)                    ***
C     *** NDS(I)....NO. DEFENSIVE SORTIES FROM BASE (I) ***
C     *** NES(I)....NO. ESCORT SORTIES FROM BASE (I)  ***
C     *** NFEBAS....NO. SUPPRESSION SORTIES AGAINST   ***
C     ***           FEBA BARRIER SAMS                ***
C     *** NINTDS...NO. SUPPRESSION SORTIES AGAINST    ***
C     ***           AREA DEPLOYED SAMS               ***
C     *** NIS(I)....NO. INTD SORTIES FROM BASE (I)    ***
C     *** NLOSE(I).....TOTAL NO.SORTIES LOST DUE TO BASE ***
C     ***           (I) DEGRADATION                   ***
C     *** NLOST(I,K)..TOTAL NO. SORTIES OF TYPE I LOST ***
C     ***           FROM BASE K                       ***
C     *** NRS(I)....NO. ESCORT SORTIES AGAINST BASE (I) ***
C     *** TFEBA.....CUMULATIVE MOVEMENT OF THE FEBA   ***
C
C
C
      PROGRAM STAG(INPUT,OUTPUT,TAPE6=OUTPUT)
      INTEGER BFORCE,RFORCE,DAY,RLOSS,BLOSS
      COMMON/A/ RSAM(2,6),BSAM(2,6),RSTAT(3),BSTAT(3),DAY
      COMMON/ESORT/ NBREF,NBIEF,NRREF,NRIEF,TFEBA,CRATIO
      COMMON/MAIN/ BFORCE,RFORCE,BLOSS,RLOSS,BARMY(3),RARMY(3),
     1BA,RA
      COMMON/FIN/ NBACLS,NRACLS
      DIMENSION BBASE(5),RBASE(5),BTYPE(3),RTYPE(3),MISSN(9),
     1BFORCE(3,9,5),RFORCE(3,9,5),IBFORCE(3,9,5),IRFORCE(3,9,5),
     2ICOLB(3,7),ICOLR(3,7)
      DIMENSION BITORY(3),RITORY(3),BBF(3),RBF(3),IWT(2),AVG(5)
      DIMENSION BLOSS(3,9,5),RLOSS(3,9,5),NRSORT(3)
      DIMENSION NBIF(3),NBRF(3),NRIF(3),NRRF(3),NBSORT(3)
      DATA (MISSN(M),M=1,9)/"TOT","ABA","RECCE","INTD","CAP",
     1"CAS","AIRDEF","DEFSP","ESCORT"/
```

```
      DATA BTYPE/"A10","F4","F111"/
      DATA RTYPE/"SU7","MIG21","MIG23"/
      DATA BLUE/"BLUE"/
      DATA RED/"RED"/
      DATA BSLO,RSLO/1.,1./
      DATA BITORY/3*250./
      DATA RITORY/3*232./
      DATA NBSORT/220,220,220/
      DATA NRSORT/220,220,220/
      DATA BSTAT/3*1./
      DATA RSTAT/3*1./
      DATA AVG/7.25,14.92,23.08,31.76,40.98/
      DATA IWT/2*50/
      DATA BSAM/14.,10.,.2,.25,.4,.5,1.,2.,3.,8.,1.,1./
      DATA RSAM/16.,12.,.2,.3,.4,.6,1.,2.,4.,10.,1.,1./
      DATA BBASE/"TOT","BB1","BB2","BB3","BARMY"/
      DATA RBASE/"TOT","RB1","RB2","RB3","RARMY"/
      DATA(((IBFORCE(I,M,K),K=1,5),M=1,9),I=1,3)/128,48,32,48,0,
     110*-1,5*0,5*-1,0,19*-1,384,3*128,5*0,-1,11*0,4*-1,0,4*-1,
     24*0,-1,10*0,96,3*32,5*0,-1,5*-1,5*0,5*-1,0,19*-1/
      DATA(((IRFORCE(I,M,K),K=1,5),M=1,9),I=1,3)/112,32,48,32,5*0,
     16*-1,5*0,5*-1,0,9*-1,10*0,120,3*40,5*0,-1,11*0,4*-1,0,4*-1,
     24*0,11*-1,328,2*104,120,5*0,-1,11*0,4*-1,0,4*-1,4*0,-1,
     310*0/
      DATA((ICOLB(I,K),K=1,7),I=1,3)/6,2,3,5,7,8,9,2,5,6,4*0,6,
     13,5,6,7,8,9/
      DATA((ICOLR(I,K),K=1,7),I=1,3)/4,3,5,6,7,0,0,4,5,6,8,9,0,
     10,2,5,6,0,0,0,0/
      DATA BARMY/3.,3.,360./
      DATA RARMY/9.,9.,1080./
C
C   *** SET DAY EQUAL 1***
C
      DAY=1
      NBACLS=0
C
C    *** SET RANDOM NUMBER GENERATOR ***
C    *** SEED WITH CLOCK TIME ***
C
      NRACLS=0
      CALL  TIME(SEED)
 1300 FORMAT(1H1,"SEED= ",A10)
      WRITE(6,1300) SEED
      CALL RANSET(SEED)
      IFLAG=0
      WRITE(6,121)
  121 FORMAT(5X,"ENTER NUMBER OF DAYS THAT YOU WISH TO PLAY"
     1"(MINIMUM OF 2, MAXIMUM OF 5)....")
      READ*,IDAY
```

```
      WRITE(6,122)
  122 FORMAT(5X,"PLAYERS SCORES ARE BASED ON THE FOLLOWING "
     1 "WEIGHTS:",/15X,"FEBA MOVEMENT...50%",/,15X,
     2"A/C LOSS RATIO..50%")
  52  WRITE(6,123)
  123 FORMAT(5X,"DO YOU WANT DIFFERENT WEIGHT FACTORS?? TYPE ...1"
     1"FOR YES...TYPE....0 FOR NO")
      READ*,IDECIDE
      IF(IDECIDE.EQ.0) GO TO 41
      IF(IDECIDE.EQ.1) GO TO 51
      WRITE(6,25)
      GO TO 52
  51  WRITE(6,124)
  124 FORMAT(5X,"ENTER THE WEIGHT OF FEBA MOVEMENT (INTEGER "
     1"BETWEEN 0 AND 100)")
      READ*,IWT(1)
      IWT(2)=100-IWT(1)
      IF(IWT(1).GE.0.AND.IWT(1).LE.100) GO TO 41
      WRITE(6,26)
  26  FORMAT(5X,"INPUT VALUE UNREASONABLE--TRY AGAIN")
      GO TO 51
  41  WRITE(6,125)
  125 FORMAT(5X,"ENTER 1 FOR STANDARD FORCE INPUT",/,5X,"ENTER 2"
     1" FOR FORCE INPUT BY PLAYER")
  25  FORMAT(2X,"INCORRECT RESPONSE--TRY AGAIN")
C
C    *** CHOOSE STANDARD FORCE INPUT ***
C
      READ*,MODE
      IF(MODE.EQ.1) GO TO 63
      IF(MODE.EQ.2) GO TO 45
      WRITE(6,25)
      GO TO 41
C
C    *** READ IN STANDARD FORCE INPUTS ***
C
  63  DO 70 I=1,3
      DO 70 M=1,9
      DO 70 K=1,5
      BFORCE(I,M,K)=IBFORCE(I,M,K)
      RFORCE(I,M,K)=IRFORCE(I,M,K)
      BLOSS(I,M,K)=0
      R[illegible](I,M,K)=0
      [illegible]
      [illegible]S(BLUE,BBASE,BTYPE,BFORCE)
      [illegible]
      [illegible]
      [illegible]ED,RBASE,RTYPE,RFORCE)
```

```
      READ*,LETSGO
   68 WRITE(6,888) BLUE
  888 FORMAT(1X,"DOES ",A5," WISH TO MOVE A/C FROM ONE BASE"
     1"TO ANOTHER??  TYPE 1 FOR YES...TYPE 0 FOR NO....")
      READ*,MOVE
      IF(MOVE.EQ.0) GO TO 95
      IF(MOVE.EQ.1) GO TO 69
      WRITE(6,25)
      GO TO 68
C
C     *** INPUT BLUE PLAYERS FORCE LIST ***
C
   69 CALL LOAD(BTYPE,BFORCE,BBASE,1,BLUE)
      CALL STATUS(BLUE,BBASE,BTYPE,BFORCE)
      GO TO 68
   95 WRITE(6,888) RED
C
C     *** DECIDE ON MOVING AIRCRAFT ***
C
      READ*,MOVE
      IF(MOVE.EQ.0) GO TO 65
      IF(MOVE.EQ.1) GO TO 13
      WRITE(6,25)
      GO TO 95
C
C     *** INPUT RED PLAYERS FORCE LIST ***
C
   13 CALL LOAD(RTYPE,RFORCE,RBASE,1,RED)
      CALL STATUS(RED,RBASE,RTYPE,RFORCE)
      GO TO 95
   45 CONTINUE
C
C     *** INITIALIZE FORCE ARRAYS ***
C
      DO 10 I=1,3
      DO 10 M=1,9
      DO 10 K=1,5
      BFORCE(I,M,K)=0
      RFORCE(I,M,K)=0
      RLOSS(I,M,K)=0
      BLOSS(I,M,K)=0
   10 CONTINUE
      DO 20 I=1,3
      J=ICOLB(I,1)
      DO 21 IJ=1,J
      NC=ICOLB(I,IJ)
      DO 22 K=1,5
   22 BFORCE(I,NC,K)=-1
   21 CONTINUE
```

```
   20  CONTINUE
       BFORCE(1,6,1)=0
       BFORCE(2,2,5)=-1
       BFORCE(2,7,5)=-1
       BFORCE(2,5,1)=0
       BFORCE(2,6,1)=0
       BFORCE(3,2,5)=-1
       BFORCE(3,6,1)=0
       DO 30 I=1,3
       JR=ICOLR(I,1)
       DO 31 IJ=1,JR
       NCR=ICOLR(I,IJ)
       DO 32 K=1,5
   32  RFORCE(I,NCR,K)=-1
   31  CONTINUE
   30  CONTINUE
       RFORCE(1,2,5)=-1
       RFORCE(1,6,1)=0
       RFORCE(2,2,5)=-1
       RFORCE(2,5,1)=0
       RFORCE(2,6,1)=0
       RFORCE(2,7,5)=-1
       RFORCE(3,2,5)=-1
       RFORCE(3,5,1)=0
       RFORCE(3,6,1)=0
       RFORCE(3,7,5)=-1
       IF(IFLAG.EQ.0) WRITE(6,700) BLUE
       CALL LOAD(BTYPE,BFORCE,BBASE,IFLAG,BLUE)
       IF(IFLAG.EQ.0) WRITE(6,700) RED
       CALL LOAD(RTYPE,RFORCE,RBASE,IFLAC,RED)
  700  FORMAT(5X,A5," SIDE WILL INPUT FORCE NOW")
C
C      *** PRINT OUT AIRCRAFT BEDDOWN ***
C
       CALL STATUS(BLUE,BBASE,BTYPE,BFORCE)
       WRITE(6,49)
   49  FORMAT(1H0,"ENTER 1 TO CONTINUE",/,/)
       READ*,LETSGO
       CALL STATUS(RED,RBASE,RTYPE,RFORCE)
       GO TO 68
   65  IFLAG=1
C
C      *** ALLOCATE BLUE AND RED SORTIES ***
C
       CALL FRAG(BLUE,BTYPE,BFORCE,RBASE,NBSORT)
       WRITE(6,49)
       READ*,LETSGO
       CALL FRAG(RED,RTYPE,RFORCE,BBASE,NRSORT)
C
```

```
C     *** AIR BASE ATTACK SUBROUTINE ***
C
      CALL ABA(BFORCE,RFORCE,1,RSAM,BLUE,RLOSS,BLOSS,NBIF,NBRF)
      WRITE(6,49)
      READ*,LETSGO
      CALL ABA(RFORCE,BFORCE,2,BSAM,RED,BLOSS,RLOSS,NRIF,NRRF)
C
C     *** GROUND WAR SUBROUTINE ***
C
      CALL GNDWAR(BSLO,RSLO)
C
C     *** PRINT OUT DAILY SUMMARY ***
C
      CALL RECAP(1,BFORCE,BLOSS,BLUE,NRIEF,BA,NRIF,BSLO,BBF,
     1BITORY,BTYPE,BARMY,BBASE,NBSORT)
      WRITE(6,49)
      READ*,LETSGO
      CALL RECAP(2,RFORCE,RLOSS,RED,NBIEF,RA,NBIF,RSLO,RBF,
     1RITORY,RTYPE,RARMY,RBASE,NRSORT)
C
C     *** CHECK FOR LAST DAY OF WAR ***
C
      IF(DAY.LE.IDAY) GO TO 71
C
C     *** COMPUTE RED AND BLUE SCORES ***
C
      HELP=TFEBA/AVG(IDAY)
      HELP1=1./HELP
      RGPTS=IWT(1)*HELP
      BGPTS=IWT(1)*HELP1
      NTRAC=560+((IDAY-1)*96)
      HELP2=FLOAT(NRACLS)/FLOAT(NTRAC)
      NTBAC=608+((IDAY-1)*105)
      HELP3=FLOAT(NBACLS)/FLOAT(NTBAC)
      HELP4=HELP3/HELP2
      HELP5=1./HELP4
      RAPTS=IWT(2)*HELP4
      BAPTS=IWT(2)*HELP5
      RSCORE=RGPTS+RAPTS
      BSCORE=BGPTS+BAPTS
      WRITE(6,127)
      WRITE(6,126) IDAY,BSCORE,RSCORE
  126 FORMAT(7X,"GAME OVER ON DAY ",I2," FINAL RESULTS:","
     1"/,16X,"BLUE...",F6.2,/,16X,"RED....",F6.2)
  127 FORMAT(1H0,50(1HX))
      WRITE(6,128)
  128 FORMAT(1H0,"***THANK YOU FOR PLAYING STAG!!***")
      STOP
      END
```

```
C
C     *** THIS SUBROUTINE ALLOWS PLAYERS ***
C     *** TO INPUT THEIR OWN INITIAL FORCES ***
C
      SUBROUTINE LOAD(TYPE,FORCE,BASE,IFLAG,SIDE)
      INTEGER FORCE
      DIMENSION TYPE(3),FORCE(3,9,5),BASE(5)
   20 FORMAT(2X,"INPUT VALUE UNREASONABLE--TRY AGAIN")
  701 FORMAT(5X,"ENTER TOTAL NO. OF ",A5,"A/C....")
  702 FORMAT(2X,/,1X,"ENTER NO. OF ",A5," A/C AT ",A5,"....")
  703 FORMAT(2X,A5," HAS ",I4,2X,A5," A/C")
      DO 50 I=1,3
C
C     *** IFLAG EQUALS 1 AFTER FIRST DAY ***
C
C     *** AFTER FIRST DAY, LOAD ALLOWS ***
C     *** PLAYERS TO MOVE A/C FROM ONE ***
C     *** BASE TO ANOTHER ***
C
      IF(IFLAG.EQ.1) GO TO 56
      WRITE(6,701) TYPE(I)
      READ*,FORCE(I,1,1)
      GO TO 55
   56 WRITE(6,703) SIDE,FORCE(I,1,1),TYPE(I)
   55 ISUM=0
      DO 51 K=2,4
      WRITE(6,702) TYPE(I),BASE(K)
      READ*,FORCE(I,1,K)
      ISUM=ISUM+FORCE(I,1,K)
      IX=FORCE(I,1,1)-ISUM
      IF(K.EQ.4.AND.IX.NE.0) GO TO 52
      IF(IX.GE.0) GO TO 51
   52 WRITE(6,20)
      GO TO 55
   51 CONTINUE
   50 CONTINUE
      RETURN
      END
C
C     *** LISTS A/C BY TYPE AND BASE FOR ***
C     *** EACH OF THE PLAYERS ***
C
      SUBROUTINE STATUS(SIDE,BASE,TYPE,FORCE)
      INTEGER FORCE
      DIMENSION BASE(5),TYPE(3),FORCE(3,9,5)
      DATA BLUE/"BLUE"/
      DATA RED/"RED"/
      WRITE(6,991)
  991 FORMAT(1X,50(1HX))
```

```
      WRITE(6,98) SIDE
 98   FORMAT(16X,A5," AIRCRAFT BEDDOWN")
      WRITE(6,992)
 992  FORMAT(1X,"-----------------------------------------")
      WRITE(6,99)
 99   FORMAT(5X,"BASE",10X,"TYPE",10X,"TOTAL",/,33X,"ACFT")
      WRITE(6,992)
      DO 908 K=2,4
      DO 909 I=1,3
      WRITE(6,106) BASE(K),TYPE(I),FORCE(I,1,K)
 106  FORMAT(5X,A4,10X,A5,10X,I4)
 107  FORMAT(10X,"TOTAL:",18X,I4)
 909  CONTINUE
      WRITE(6,992)
 908  CONTINUE
      WRITE(6,992)
      NSUM=FORCE(1,1,1)+FORCE(2,1,1)+FORCE(3,1,1)
      WRITE(6,107) NSUM
      WRITE(6,991)
      RETURN
      END
C
C    *** ALLOWS PLAYERS TO ALLOCATE SORTIES BY ***
C    *** TYPE A/C TO TARGETS OF THEIR CHOICE ***
C
      SUBROUTINE FRAG(SIDE,TYPE,FORCE,BASE,NSORT)
      INTEGER FORCE,SUM
      DIMENSION MISSN(9),TYPE(3),FORCE(3,9,5),BASE(5),NSORT(3)
      DIMENSION SUM(3),NLOST(3),NLOSE(3,3),NTRY(3)
      DATA MISSN/"TOT","ABA","RECCE","INTD","CAP","CAS",
     1"AIRDEF","DEFSP","ESCORT"/
      WRITE(6,141) SIDE
      WRITE(6,140) SIDE
 141  FORMAT(5X,A5," SORTIES WILL BE ALLOCATED NOW")
 140  FORMAT(5X,"ENTER NO. OF ",A5,"SORTIES BY TYPE AIRCRAFT TO "
     1"EACH MISSION(AS IT IS DISPLAYED)")
      WRITE(6,145)
      WRITE(6,148)
      WRITE(6,142)
      WRITE(6,992)
 992  FORMAT(1X,"-----------------------------------------")
 145  FORMAT(5X,"AIRCRAFT ALLOCATED TO EACH BASE (UNDER HEADING "
     1"OF...TARGET) FOR AIR DEFENSE WILL BE USED TO PROTECT THAT"
     2" AIRBASE FROM ATTACK.")
 148  FORMAT(2X,"FOR EXAMPLE,IF YOU ONLY HAVE 20 F4 A/C AT BASE"
     1" 2 YOU CANNOT ALLOCATE",/," MORE THAN 20 F4 A/C TO THE"
     2" AIR DEFENSE OF BASE 2.")
 142  FORMAT(2X,/,2X,"SIDE",5X,"TYPE A/C",3X,"MISSION",4X,"MAX "
     1"AVAIL",3X,"TARGET")
```

```
      M2=M3=M4=M5=0
      M6=M7=M8=M9=0
      NTRY(1)=NTRY(2)=NTRY(3)=0
      SUM(1)=FORCE(1,1,2)+FORCE(2,1,2)+FORCE(3,1,2)
      SUM(2)=FORCE(1,1,3)+FORCE(2,1,3)+FORCE(3,1,3)
      SUM(3)=FORCE(1,1,4)+FORCE(2,1,4)+FORCE(3,1,4)
      DO 10 I=1,3
C
C    *** NLOST IS SORTIES LOST DUE TO DEGRADED ***
C    *** BASE CAPABILITY--EITHER LOGISTICS ***
C    *** SHORTAGES OR AIRBASE DAMAGE ***
C
      NLOST(I)=AMINO(NSORT(I)-SUM(I),0)
      IF(NLOST(I).LT.0) NLOST(I)=-NLOST(I)
      DO 15 K=1,3
      N=K+1
      IF(SUM(I).EQ.0) SUM(I)=1
C
C    *** NLOSE IS SORTIES OF TYPE I LOST ***
C    *** FROM BASE K ***
C
      NLOSE(I,K)=NLOST(I)*FORCE(I,1,N)/SUM(I)
   15 CONTINUE
   10 CONTINUE
      DO 20 I=1,3
      DO 25 K=1,3
      NTRY(I)=NTRY(I)+NLOSE(I,K)
   25 CONTINUE
   20 CONTINUE
      DO 139 I=1,3
      NAC=FORCE(I,1,1)-NTRY(I)
      DO 149 M=2,9
      NTOT=0
      DO 159 K=1,5
      IF(FORCE(I,M,K).EQ.-1) GO TO 159
  144 IF(K.EQ.1) WRITE(6,143) SIDE,TYPE(I),MISSN(M),NAC,BASE(K)
      IF(K.GE.2) WRITE(6,143) SIDE,TYPE(I),MISSN(M),NTOT,BASE(K)
  143 FORMAT(2X,A4,6X,A5,6X,A6,6X,I3,8X,A5,"....")
      READ*,FORCE(I,M,K)
      IF(K.EQ.1)NTOT=FORCE(I,M,K)
C
C    *** CAN'T ALLOCATE MORE THAN YOU HAVE ***
C
      IF(NTOT.GT.NAC) GO TO 49
      IF(M.EQ.5.OR.M.EQ.6) NAC=NAC-FORCE(I,M,K)
      IF(NAC.LT.0) GO TO 39
      IF(NTOT.EQ.0) GO TO 169
      IF(K.GE.2) NTOT=NTOT-FORCE(I,M,K)
      IF(NTOT.LT.0) GO TO 29
```

```
      IF(FORCE(I,7,2).GT.FORCE(I,1,2)) GO TO 29
      IF(FORCE(I,7,3).GT.FORCE(I,1,3)) GO TO 29
      IF(FORCE(I,7,4).GT.FORCE(I,1,4)) GO TO 29
      IF(NTOT.GE.0) GO TO 146
  150 IF(FORCE(I,M,K).GE.0.AND.FORCE(I,M,K).LE.NAC) GO TO 146
  29  PRINT 21
  21  FORMAT(2X,"INPUT VALUE UNREASONABLE--TRY AGAIN")
      NTOT=NTOT+FORCE(I,M,K)
      GO TO 144
  39  PRINT 21
      NAC=NAC+FORCE(I,M,K)
      GO TO 144
  49  PRINT 21
      GO TO 144
  146 IF(K.GE.2) NAC=NAC-FORCE(I,M,K)
C
C     *** M IS COUNTER ON NO. OF SORTIES ALLOCATED ***
C     *** TO EACH MISSION ***
C
  169 IF(M.EQ.2.AND.K.GE.2) M2=M2+FORCE(I,M,K)
      IF(M.EQ.3.AND.K.GE.2) M3=M3+FORCE(I,M,K)
      IF(M.EQ.4.AND.K.GE.2) M4=M4+FORCE(I,M,K)
      IF(M.EQ.5) M5=M5+FORCE(I,M,K)
      IF(M.EQ.6) M6=M6+FORCE(I,M,K)
      IF(M.EQ.7.AND.K.GE.2) M7=M7+FORCE(I,M,K)
      IF(M.EQ.8.AND.K.GE.2) M8=M8+FORCE(I,M,K)
      IF(M.EQ.9.AND.K.GE.2) M9=M9+FORCE(I,M,K)
      IF(NAC.EQ.0) GO TO 139
      IF(NTOT.EQ.0) GO TO 149
  159 CONTINUE
  149 CONTINUE
  139 CONTINUE
      NSRT=M2+M3+M4+M5+M6+M7+M8+M9
      IF(NSRT.EQ.0) GO TO 11
C
C     *** P IS PERCENTAGE OF SORTIES ALLOCATED ***
C     *** TO EACH MISSION ***
C
      P2=FLOAT(M2)/NSRT
      P3=FLOAT(M3)/NSRT
      P4=FLOAT(M4)/NSRT
      P5=FLOAT(M5)/NSRT
      P6=FLOAT(M6)/NSRT
      P7=FLOAT(M7)/NSRT
      P8=FLOAT(M8)/NSRT
      P9=FLOAT(M9)/NSRT
      GO TO 12
  11  P2=P3=P4=P5=P6=P7=P8=P9=0.
  12  P9T=P2+P3+P4+P5+P6+P7+P8+P9
```

```
      WRITE(6,137) (MISSN(M),M=2,9)
      WRITE(6,138) M2,M3,M4,M5,M6,M7,M8,M9,NSRT
      WRITE(6,199) P2,P3,P4,P5,P6,P7,P8,P9,P9T
  137 FORMAT(1X,"MISSN:",3X,A6,A8,A7,2(A6),A9,A8,A9,"TOT")
  138 FORMAT(10X,I3,4X,I3,5X,I3,2(3X,I3),2(5X,I3),6X,I3,4X,I3)
  199 FORMAT(1X,"PERCENT",F5.3,2X,F5.3,3X,F5.3,2(1X,F5.3),
     12(3X,F5.3),4X,F5.3,2X,F5.3)
      RETURN
      END
C
C     *** SAM SUPPRESSION COMPUTES LOSSES TO ***
C     *** SUPPRESSION A/C AND REDUCES NO. OF ***
C     *** SAM SITES AVAILABLE TO KILL OTHER A/C ***
C
      SUBROUTINE SAMSUP(SAM,FORCE1,FORCE2,ALOSS,DLOSS)
      INTEGER FORCE1,FORCE2,ALOSS,DLOSS
      COMMON/A/ RSAM(2,6),BSAM(2,6),RSTAT(3),BSTAT(3)
      COMMON/B/ ENSS,ENMS
      DIMENSION FORCE1(3,9,5),FORCE2(3,9,5),SAM(2,6)
      DIMENSION ALOSS(3,9,5),DLOSS(3,9,5)
      NINTDS=0
      NFEBAS=0
      DO 15 I=1,3
      IF(FORCE1(I,8,5).LE.0) GO TO 15
      NFEBAS=NFEBAS+FORCE1(I,8,5)
   15 CONTINUE
      PSFAAA=.95
      RN1=RANF(D)
      IF(RN1.GT..25) PSFAAA=.96
      IF(RN1.GT..50) PSFAAA=.97
      IF(RN1.GT..75)PSFAAA=.98
      NDAC=0
      DO 18 I=1,3
      IF(FORCE2(I,5,1).LE.0) GO TO 18
      NDAC=NDAC+FORCE2(I,5,1)
   18 CONTINUE
      DAP=.8*NDAC
      PNED=EXP(-DAP/100)
      NFEBA=NFEBAS*PSFAAA
      NSKFBA=NFEBAS-NFEBA
      FRAC=1.-EXP((-SAM(1,1)*2.*SAM(1,5))/100.)
      ENSS=FRAC*SAM(1,3)*SAM(1,4)*.25
      NSACSF=NFEBA*(1.-SAM(1,2))**ENSS
      NSKCAP=NFEBA-NSACSF
      IF(SAM(1,1).GT.0.) GO TO 20
      NSSUPF=0
      GO TO 21
   20 NSSUPF=SAM(1,1)*(1.-EXP((FLOAT(-NSACSF)*.25)/SAM(1,1)))
   21 NSDESF=.3*NSSUPF
```

```
C
C     *** COMPUTES NO. OF SAM SITES SUPPRESSED ***
C     *** THESE SITES ARE AT THE FEBA ***
C
      SAM(1,6)=1.-(NSSUPF/SAM(1,1))
      SAM(1,1)=SAM(1,1)-NSDESF
      PSIAAA=.95
      RN2=RANF(D)
      IF(RN2.GT..25)PSIAAA=.96
      IF(RN2.GT..50)PSIAAA=.97
      IF(RN2.GT..75)PSIAAA=.98
      DO 19 I=1,3
      IF(FORCE1(I,8,1).LE.0) GO TO 19
      NSUM=FORCE1(I,8,1)-FORCE1(I,8,5)
      NINTDS=NINTDS+NSUM
   19 CONTINUE
      NINTD=NINTDS*PSIAAA
      NSKAAA=NINTDS-NINTD
      FRAC=1.-EXP((-SAM(2,1)*2.*SAM(2,5))/100.)
      ENMS=FRAC*SAM(2,3)*SAM(2,4)*.25
      NSACSI=NINTD*(1.-SAM(2,2))**ENMS
      NSKI=NINTD-NSACSI
      IF(SAM(2,1).GT.0.) GO TO 30
      NSSUPI=0
      GO TO 31
   30 NSSUPI=SAM(2,1)*(1.-EXP((FLOAT(-NSACSI)*.25)/SAM(2,1)))
   31 SDESI=FLOAT(NSSUPI)*.3
C
C     *** COMPUTES NO. OF SAM SITES SUPPRESSED ***
C     *** THESE ARE AREA DEPLOYED SAMS ***
C
      SAM(2,6)=1.-(NSSUPI/SAM(2,1))
      SAM(2,1)=SAM(2,1)-SDESI
      NSK=NSKFBA+NSKCAP+NSKAAA+NSKI
      NTOTS=NFEBAS+NINTDS
      NSR=NTOTS-NSK
      IF(NTOTS.LE.0) GO TO 173
      RATIO=FLOAT(FORCE1(1,8,1))/FLOAT(NTOTS)
      JF=NSK
C     *** UPDATES LOSS ARRAYS WITH NO. A/C KILLED ***
C     *** LOSSES DISTRIBUTED IN PROPORTION TO NO. ***
C     *** OF TYPE A/C PERFORMING MISSION ***
C
      DO 170 I=1,JF
      RN3=RANF(D)
      IF(RN3.GT.RATIO) GO TO 171
      ALOSS(1,8,1)=ALOSS(1,8,1)+1
      GO TO 170
  171 RATIOM=FLOAT(FORCE1(1,8,1)+FORCE1(2,8,1))/FLOAT(NTOTS)
```

```
      IF(RN3.GT.RATIOM) GO TO 172
      ALOSS(2,8,1)=ALOSS(2,8,1)+1
      GO TO 170
  172 ALOSS(3,8,1)=ALOSS(3,8,1)+1
  170 CONTINUE
  173 WRITE(6,180)
      WRITE(6,182)
  182 FORMAT(1X,23(1HX))
  180 FORMAT(1H0,"SAM SUPPRESSION MISSION")
      WRITE(6,181) NTOTS,NSK,NSR
  181 FORMAT(1X,"TOTAL A/C ALLOCATED...",I3,/," A/C LOST..."
     1"..........",I3," A/C REMAINING.........",I3,/)
      RETURN
      END
C
C    *** ABA COMPUTES DAMAGE TO AIRBASES AND LOSSES ***
C    *** DUE TO SAMS, AAA, AND AIR DEFENSE ***
C
      SUBROUTINE ABA(ATTACK,DEFEND,NCHECK,SAM,SIDE,DLOSS,ALOSS,
     1NIF,NRF)
      INTEGER ATTACK,DEFEND,DLOSS,ALOSS
      COMMON/A/ RSAM(2,6),BSAM(2,6),RSTAT(3),BSTAT(3)
      COMMON/B/ ENSS,ENMS
      DIMENSION ATTACK(3,9,5),DEFEND(3,9,5),NAS(3),NDS(3),NES(3),
     1SAM(2,6),DLOSS(3,9,5),ALOSS(3,9,5),PK(2,2),NASSA(3),
     2NESSA(3),NASSS(3),NESSS(3),NEK(3),NAK(3),NDK(3),
     3NASS(3),NESS(3),NBEFF(3),NREFF(3),RAKF(3),BAKF(3)
      DIMENSION FACT(5),NIS(3),NRS(3),NIK(3),NRK(3),NIF(3)
      DIMENSION NAKA(3),NIKA(3),NRKA(3),NEKA(3),NRF(3)
      DATA PK/.1,.1,.2,.15/
      DATA FACT/.0015,.001,.002,.002,.003/
      WRITE(6,175)
      WRITE(6,176)
      WRITE(6,150) SIDE
      WRITE(6,176)
      WRITE(6,175)
C
C    *** SAM SUPPRESSION RESULTS COMPUTED HERE ***
C
      IF(NCHECK.EQ.1) CALL SAMSUP(RSAM,ATTACK,DEFEND,ALOSS,DLOSS)
      IF(NCHECK.EQ.2) CALL SAMSUP(BSAM,ATTACK,DEFEND,ALOSS,DLOSS)
      IA=1
      ID=2
      IF(NCHECK.EQ.2) IA=2
      IF(NCHECK.EQ.2) ID=1
      NAS(1)=NAS(2)=NAS(3)=0
      NES(1)=NES(2)=NES(3)=0
      NDS(1)=NDS(2)=NDS(3)=0
      NIS(1)=NIS(2)=NIS(3)=0
```

```
      NRS(1)=NRS(2)=NRS(3)=0
      NIF(1)=NIF(2)=NIF(3)=0
      NRF(1)=NRF(2)=NRF(3)=0
C
C    *** COMPUTES NO. A/C BY TYPE I ATTACKING ***
C    *** TARGET K ***
C    *** ALSO COMPUTES NO. OF ESCORT, RECCE, ***
C    *** INTERDICTION, AND DEFENSIVE SORTIES HERE ***
C
      DO 110 K=1,3
      J=K+1
      DO 115 I=1,3
      IF(ATTACK(I,2,J).GT.0) NAS(K)=NAS(K)+ATTACK(I,2,J)
      IF(ATTACK(I,9,J).GT.0) NES(K)=NES(K)+ATTACK(I,9,J)
      IF(DEFEND(I,7,J).GT.0) NDS(K)=NDS(K)+DEFEND(I,7,J)
      IF(ATTACK(I,4,J).GT.0) NIS(K)=NIS(K)+ATTACK(I,4,J)
      IF(ATTACK(I,3,J).GT.0) NRS(K)=NRS(K)+ATTACK(I,3,J)
  115 CONTINUE
  110 CONTINUE
  175 FORMAT(23X,"**********************")
  176 FORMAT(23X,"*",20X,"*")
  150 FORMAT(23X,"*",2X,A4," AIR RESULTS",2X,"*")
      WRITE(6,151)
  151 FORMAT(10X,"MISSION",3X,"A/C ALLOCATED",3X,"G-A LOSS",
     13X,"A-A LOSS",3X,"A/C REMAINING")
      DO 120 J=1,3
      PROB=.95
      RN1=RANF(D)
      IF(RN1.GT..25) PROB=.96
      IF(RN1.GT..50) PROB=.97
      IF(RN1.GT..75) PROB=.98
C
C    *** LOSSES DUE TO AAA ***
C
      NASSA(J)=NAS(J)*PROB
      NESSA(J)=NES(J)*PROB
      NISSA=NIS(J)*PROB
      NRSSA=NRS(J)*PROB
      NAKAAA=NAS(J)-NASSA(J)
      NEKAAA=NES(J)-NESSA(J)
      NIKAAA=NIS(J)-NISSA
      NRKAAA=NRS(J)-NRSSA
      ENSA=ENSS*2.*SAM(1,6)
      ENMA=ENMS*2.*SAM(2,6)
C
C    *** LOSSES DUE TO SAMS ***
C
      NASSS(J)=NASSA(J)*(1.-SAM(1,2))**ENSA
      NASS(J)=NASSS(J)*(1.-SAM(2,2))**ENMA
```

```
      NESSS(J)=NESSA(J)*(1.-SAM(1,2))**ENSA
      NESS(J)=NESSS(J)*(1.-SAM(2,2))**ENMA
      NISSS=NISSA*(1.-SAM(1,2))**ENSA
      NISS=NISSS*(1.-SAM(2,2))**ENMA
      NRSSS=NRSSA*(1.-SAM(1,2))**ENSA
      NRSS=NRSSS*(1.-SAM(2,2))**ENMA
      NAKSAM=NASSA(J)-NASS(J)
      NEKSAM=NESSA(J)-NESS(J)
      NIKSAM=NISSA-NISS
      NRKSAM=NRSSA-NRSS
      IF(NESS(J).LE.0) GO TO 99
      ND=AMINO(NESS(J),NDS(J))
      IF(ND.LE.0) GO TO 99
C
C    *** COMPUTE ESCORT LOSSES TO AIR DEFENSE ***
C    *** COMPUTE DEFENDER LOSSES TO ESCORTS ***
C
      FRAC=NESS(J)/ND
      PEK=1.-(1.-PK(ID,2))**FRAC
      PDK=1.-(1.-PK(IA,2))**(1./FRAC)
      CALL RANDOM(NESS(J),PEK,NEKA(J))
      NEK(J)=NEKA(J)+NEKAAA+NEKSAM
      CALL RANDOM(ND,PDK,NDKE)
      NDSE=AMAX0(0,NDS(J)-ND)
      GO TO 100
   99 NEKA(J)=0
C
C    *** COMPUTE ATTACKER LOSSES TO AIR DEFENSE ***
C    *** COMPUTE DEFENDER LOSSES TO ATTACKERS ***
C
      NEK(J)=NEKA(J)+NEKAAA+NEKSAM
      NDKE=0
      NDSE=NDS(J)
  100 DEP=FLOAT(NDSE)*.6
      DEP=AMAX1(0.0,DEP)
      IF(NASS(J).GT.0) GO TO 60
      PADE=0.
      NADE=0
      PAK=0.
      GO TO 61
   60 PADE=1.-EXP(-DEP/FLOAT(NASS(J)))
      NADE=FLOAT(NASS(J))*PADE
      IF(NADE.GT.0) GO TO 62
      PAK=0.
      GO TO 61
   62 PAK=1.-EXP((-PK(ID,2)*DEP)/NADE)
   61 CALL RANDOM(NADE,PAK,NAKA(J))
      NAK(J)=NAKA(J)+NAKAAA+NAKSAM
      IF(DEP.GT.0.) GO TO 70
```

```
      PDK=0.
      GO TO 71
   70 PDK=1.-EXP((-PK(IA,1)*NADE)/DEP)
   71 NEP=DEP
      CALL RANDOM(NEP,PDK,NDKA)
      IF(NISS.GT.0) GO TO 300
      NIDE=0
      PIDE=0.
      PIK=0.
      GO TO 301
C
C     *** COMPUTE INTD LOSSES TO AIR DEFENSE ***
C     *** COMPUTE DEFENDER LOSSES TO INTD ***
C
  300 PIDE=1.-EXP((-.9*DEP)/FLOAT(NISS))
      NIDE=FLOAT(NISS)*PIDE
      PIK=1.-EXP((-PK(ID,2)*DEP*.9)/NISS)
  301 CALL RANDOM(NIDE,PIK,NIKA(J))
      NIK(J)=NIKA(J)+NIKAAA+NIKSAM
      NIF(J)=NIS(J)-NIKAAA-NIKSAM-NIDE
      IF(DEP.GT.0) GO TO 310
      PDK=0.
      GO TO 311
C
C     *** COMPUTE RECCE LOSSES TO AIR DEFENSE ***
C
  310 PDK=1.-EXP((-PK(IA,1)*NIDE)/(.9*DEP))
  311 NEP=.9*DEP
      CALL RANDOM(NEP,PDK,NDKI)
      NDK(J)=NDKA+NDKI+NDKE
      IF(NRSS.GT.0) GO TO 400
      NRDE=0
      PRDE=0.
      PRK=0.
      GO TO 401
  400 PRDE=1.-EXP((-.7*DEP)/FLOAT(NRSS))
      NRDE=FLOAT(NRSS)*PRDE
      PRK=1.-EXP((-PK(ID,2)*DEP*.7)/NRSS)
  401 CALL RANDOM(NRDE,PRK,NRKA(J))
      NRK(J)=NRKA(J)+NRKAAA+NRKSAM
      NRF(J)=NRS(J)-NRKAAA-NRKSAM-NRDE
      JX=J+1
      IF(NDS(J).LE.0) GO TO 80
      RATIO=FLOAT(DEFEND(2,7,JX))/FLOAT(NDS(J))
      JD=NDK(J)
C
C     *** UPDATE DEFENDERS LOSS ARRAY ***
C
      DO 130 I=1,JD
```

AD-A083 911 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 15/7
STAG: A TWO PERSON SIMULATED TACTICAL AIR WAR GAME.(U)
MAR 80 J M FOLEY
UNCLASSIFIED AFIT/GST/OS/80M-2 NL

2 of 2
AD
A083911

END
DATE
FILMED
6-80
DTIC

```
      RN2=RANF(D)
      IF(RN2.GT.RATIO) GO TO 131
      DLOSS(2,7,JX)=DLOSS(2,7,JX)+1
      GO TO 130
  131 DLOSS(3,7,JX)=DLOSS(3,7,JX)+1
  130 CONTINUE
  80  JY=J+1
      IF(NAS(J).LE.0) GO TO 90
      RATIO=FLOAT(ATTACK(2,2,JY))/FLOAT(NAS(J))
      JE=NAK(J)
C
C    *** UPDATE ATTACKERS LOSS ARRAY ***
C
      DO 140 I=1,JE
      RN3=RANF(D)
      IF(RN3.GT.RATIO) GO TO 141
      ALOSS(2,2,JY)=ALOSS(2,2,JY)+1
      GO TO 140
  141 RATIOM=FLOAT(ATTACK(3,2,JY)+ATTACK(2,2,JY))/FLOAT(NAS(J))
      IF(RN3.GT.RATIOM) GO TO 142
      ALOSS(3,2,JY)=ALOSS(3,2,JY)+1
      GO TO 140
  142 ALOSS(1,2,JY)=ALOSS(1,2,JY)+1
  140 CONTINUE
  90  JZ=J+1
      IF(NES(J).LE.0) GO TO 109
      RATIO=FLOAT(ATTACK(1,9,JZ))/FLOAT(NES(J))
      JF=NEK(J)
C
C    *** UPDATE ESCORT'S LOSS ARRAY ***
C
      DO 160 I=1,JF
      RN4=RANF(D)
      IF(RN4.GT.RATIO) GO TO 161
      ALOSS(1,9,JZ)=ALOSS(1,9,JZ)+1
      GO TO 160
  161 RATIOM=FLOAT(ATTACK(1,9,JZ)+ATTACK(2,9,JZ))/FLOAT(NES(J))
      IF(RN4.GT.RATIOM) GO TO 162
      ALOSS(2,9,JZ)=ALOSS(2,9,JZ)+1
      GO TO 160
  162 ALOSS(3,9,JZ)=ALOSS(3,9,JZ)+1
  160 CONTINUE
  109 JA=J+1
      IF(NIS(J).LE.0) GO TO 500
      RATIO=FLOAT(ATTACK(1,4,JA))/FLOAT(NIS(J))
      JM=NIK(J)
C
C    *** UPDATE INTD LOSS ARRAY ***
C
```

```
      DO 501 I=1,JM
      RN=RANF(D)
      IF(RN.GT.RATIO) GO TO 502
      ALOSS(1,4,JA)=ALOSS(1,4,JA)+1
      GO TO 501
  502 RATIO2=FLOAT(ATTACK(1,4,JA)+ATTACK(2,4,JA))/FLOAT(NIS(J))
      IF(RN.GT.RATIO2) GO TO 503
      ALOSS(2,4,JA)=ALOSS(2,4,JA)+1
      GO TO 501
  503 ALOSS(3,4,JA)=ALOSS(3,4,JA)+1
  501 CONTINUE
  500 IF(NRS(J).LE.0) GO TO 600
      RATIO=FLOAT(ATTACK(2,3,JA))/FLOAT(NRS(J))
      JN=NRK(J)
C
C UPDATE RECCE LOSS ARRAY ***
C
      DO 601 I=1,JN
      RN=RANF(D)
      IF(RN.GT.RATIO) GO TO 602
      ALOSS(2,3,JA)=ALOSS(2,3,JA)+1
      GO TO 601
  602 ALOSS(3,3,JA)=ALOSS(3,3,JA)+1
  601 CONTINUE
C
C    *** PRINT OUT AIRBASE ATTACK RESULTS ***
C
  600 N1=NAKAAA+NAKSAM
      N2=NIKAAA+NIKSAM
      N3=NRKAAA+NRKSAM
      N4=NEKAAA+NEKSAM
      WRITE(6,152) J
  152 FORMAT(2X,"TARGET--AIRBASE NO.",I2)
      NAR=NAS(J)-NAK(J)
      WRITE(6,153) NAS(J),N1,NAKA(J),NAR
  153 FORMAT(1H0,10X,"ATTACK",8X,I4,10X,I4,7X,I4,7X,I4)
      NRR=NRS(J)-NRK(J)
      WRITE(6,154) NRS(J),N3,NRKA(J),NRR
  154 FORMAT(1H0,10X,"RECCE",9X,I4,10X,I4,7X,I4,7X,I4)
      NIR=NIS(J)-NIK(J)
      WRITE(6,158) NIS(J),N2,NIKA(J),NIR
  158 FORMAT(1H0,10X,"INTDXN",8X,I4,10X,I4,7X,I4,7X,I4)
      NER=NES(J)-NEK(J)
      WRITE(6,155) NES(J),N4,NEKA(J),NER
  155 FORMAT(1H0,10X,"ESCORT",8X,I4,10X,I4,7X,I4,7X,I4)
      IF(NCHECK.EQ.1) NBEFF(J)=NAS(J)-NAKAAA-NAKSAM-NADE
      IF(NCHECK.EQ.2) NREFF(J)=NAS(J)-NAKAAA-NAKSAM-NADE
      IF(NAS(J).LE.0) GO TO 119
      JA=J+1
```

```
      IF(NCHECK.EQ.1) GO TO 860
      SUM1=(ATTACK(1,2,JA)*FACT(1))+(ATTACK(2,2,JA)*FACT(2))
     1+(ATTACK(3,2,JA)*FACT(3))
      RAKF(J)=(SUM1*NREFF(J))/NAS(J)
      BSTAT(J)=BSTAT(J)-RAKF(J)
      IF(NRR.GT.0) WRITE(6,156) BSTAT(J)
      IF(NRR.LE.0) WRITE(6,157)
      GO TO 120
  860 SUM2=(ATTACK(2,2,JA)*FACT(4))+(ATTACK(3,2,JA)*FACT(5))
      BAKF(J)=(SUM2*NBEFF(J))/NAS(J)
      RSTAT(J)=RSTAT(J)-BAKF(J)
      IF(NRR.GT.0) WRITE(6,156) RSTAT(J)
      IF(NRR.LE.0) WRITE(6,157)
  156 FORMAT(5X,"TARGET STATUS IS ",F6.3)
  157 FORMAT(5X,"NO INFO ON THIS TARGET")
      GO TO 120
  119 WRITE(6,157)
  120 CONTINUE
      RETURN
      END
C
C    *** GIVEN THE NO. A/C AT RISK AND ***
C    *** THEIR PK, RANDOM DETERMINES NO. A/C KILLED ***
C
      SUBROUTINE RANDOM(N,PROB,NK)
      NK=0
      IF(N.LE.0) GO TO 200
      DO 199 I=1,N
      RN=RANF(D)
      IF(RN.GT.PROB) GO TO 199
      NK=NK+1
  199 CONTINUE
  200 RETURN
      END
C
C    *** GNDWAR COMPUTES DAILY FEBA MOVEMENT, ***
C    *** LOSSES DUE TO GROUND FORCES, CAS SORTIES, ***
C    *** CAP SORTIES, AND SLOW DOWN FACTOR DUE TO ***
C    *** LOGISTICS SHORTAGES    ***
C
      SUBROUTINE GNDWAR(BSLOW,RSLOW)
      INTEGER BFORCE,RFORCE,BLOSS,RLOSS,DAY
      COMMON/A/ RSAM(2,6),BSAM(2,6),RSTAT(3),BSTAT(3),DAY
      COMMON/ESORT/ NBREF,NBIEF,NRREF,NRIEF,TFEBA,CRATIO
      COMMON/MAIN/ BFORCE(3,9,5),RFORCE(3,9,5),BLOSS(3,9,5),
     1RLOSS(3,9,5), BARMY(3),RARMY(3),BA,RA
      DIMENSION NBCAS(3),NRCAS(3),NBSAAA(3),NBKAAA(3),NRSAAA(3),
     1NRKAAA(3),NBSS(3),NBKS(3),NRSS(3),NRKS(3),NRCAP(2),
     2CAPEP(3),NCAS(3),BCAS(3),JCAS(3),RCAS(3),BFRAC(3),RFRAC(3)
```

```
      DATA CAPEP/.8,.7,.8/
      DATA BFRAC/.0003,.00015,.0003/
      DATA RFRAC/.00015,.0001,.00015/
C
C    *** INITIALIZE VARIABLES ***
C
      NBCAS(1)=NBCAS(2)=NBCAS(3)=0
      NRCAS(1)=NRCAS(2)=NRCAS(3)=0
      BSUM=RSUM=YSUM=ZSUM=0.
      NBREC=NBITD=NBES=0
      NRREC=NRITD=NRES=0
      VMAX=30.
      X1=-0.75
      X2=8.75
      X3=1.3
      IF(DAY.EQ.1) TFEBA=0.
C
C    *** COMPUTE NO.CAS,CAP,ESCORT,RECCE,AND INTD ***
C    *** SORTIES FOR BLUE AND RED ***
C
      DO 110 I=1,3
      IF(BFORCE(I,6,1).GT.0) NBCAS(I)=NBCAS(I)+BFORCE(I,6,1)
      IF(RFORCE(I,6,1).GT.0) NRCAS(I)=NRCAS(I)+RFORCE(I,6,1)
      IF(BFORCE(I,9,5).GT.0) NBES=NBES+BFORCE(I,9,5)
      IF(BFORCE(I,3,5).GT.0) NBREC=NBREC+BFORCE(I,3,5)
      IF(BFORCE(I,4,5).GT.0) NBITD=NBITD+BFORCE(I,4,5)
      IF(RFORCE(I,9,5).GT.0) NRES=NRES+RFORCE(I,9,5)
      IF(RFORCE(I,3,5).GT.0) NRREC=NRREC+RFORCE(I,3,5)
      IF(RFORCE(I,4,5).GT.0) NRITD=NRITD+RFORCE(I,4,5)
C
C    *** COMPUTE LOSSES TO AAA ***
C
      NBSAAA(I)=NBCAS(I)*.97
      NBKAAA(I)=NBCAS(I)-NBSAAA(I)
      BLOSS(I,6,1)=BLOSS(I,6,1)+NBKAAA(I)
      NRSAAA(I)=NRCAS(I)*.97
      NRKAAA(I)=NRCAS(I)-NRSAAA(I)
      RLOSS(I,6,1)=RLOSS(I,6,1)+NRKAAA(I)
C
C    *** COMPUTE LOSSES TO SAMS ***
C
      FRAC=1.-EXP((-RSAM(1,1)*2.*RSAM(1,5))/100.)
      ENS=FRAC*RSAM(1,3)*RSAM(1,4)*.5*RSAM(1,6)
      NBSS(I)=NBSAAA(I)*(1.-RSAM(1,2))**ENS
      NBKS(I)=NBSAAA(I)-NBSS(I)
      BLOSS(I,6,1)=BLOSS(1,6,1)+NBKS(I)
      FRAC=1.-EXP((-BSAM(1,1)*2.*BSAM(1,5))/100.)
      ENSR=FRAC*BSAM(1,3)*BSAM(1,4)*.5*BSAM(1,6)
      NRSS(I)=NRSAAA(I)*(1.-BSAM(1,2))**ENSR
```

```
      NRKS(I)=NRSAAA(I)-NRSS(I)
      RLOSS(I,6,1)=RLOSS(I,6,1)+NRKS(I)
  110 CONTINUE
C
C    *** COMPUTE LOSSES TO AAA ***
C
      NBRSA=NBREC*.98
      NBISA=NBITD*.98
      NRRSA=NRREC*.98
      NRISA=NRITD*.98
      NBESSA=NBES*.98
      NRESSA=NRES*.98
C
C    *** COMPUTE LOSSES TO SAMS ***
C
      NBRSS=NBRSA*(1.-RSAM(1,2))**ENS
      NBISS=NBISA*(1.-RSAM(1,2))**ENS
      NBESSS=NBESSA*(1.-RSAM(1,2))**ENS
      NRRSS=NRRSA*(1.-BSAM(1,2))**ENSR
      NRISS=NRISA*(1.-BSAM(1,2))**ENSR
      NRESSS=NRESSA*(1.-BSAM(1,2))**ENSR
      NBCAP=BFORCE(2,5,1)-BLOSS(2,5,1)
      DO 200 I=2,3
      J=I-1
      IF(RFORCE(I,5,1).LE.0) GO  TO 201
      NRCAP(J)=RFORCE(I,5,1)-RLOSS(I,5,1)
      GO TO 200
  201 NRCAP(J)=0
  200 CONTINUE
      NTOT=NRCAP(1)+NRCAP(2)
      BEP=FLOAT(NBCAP)*CAPEP(1)
      REP=FLOAT(NRCAP(1))*CAPEP(2)+FLOAT(NRCAP(2))*CAPEP(3)
      BCASEP=(NBSS(1)*.9)+(NBSS(2)*.7)+(NBSS(3)*.95)
      RCASEP=(NRSS(1)*.65)+(NRSS(2)*.7)+(NRSS(3)*.8)
      IF(BCASEP.GT.0) GO TO 599
      NBCASK=0
      GO TO 202
C
C    *** BLUE CAS KILLED BY CAP ***
C
  599 PD=1.-EXP(-REP/BCASEP)
      ND=BCASEP*PD
      PCASK=1.-EXP((-.2*REP)/BCASEP)
      CALL RANDOM(ND,PCASK,NBCASK)
  202 NBTOT=NBSS(1)+NBSS(2)+NBSS(3)
      IF(NBTOT.LE.0) GO TO 13
      RATIO=FLOAT(NBSS(1))/FLOAT(NBTOT)
      JF=NBCASK
C
```

```
C     *** UPDATE BLUE LOSS ARRAY ***
C
      DO 400 I=1,JF
      RN=RANF(D)
      IF(RN.GT.RATIO) GO TO 401
      BLOSS(1,6,1)=BLOSS(1,6,1)+1
      GO TO 400
  401 RATIOM=FLOAT(NBSS(1)+NBSS(2))/FLOAT(NBTOT)
      IF(RN.GT.RATIOM) GO TO 402
      BLOSS(2,6,1)=BLOSS(2,6,1)+1
      GO TO 400
  402 BLOSS(3,6,1)=BLOSS(3,6,1)+1
  400 CONTINUE
   13 NDR=AMINO(NBESSS,NRCAP)
      IF(NDR.LE.0) GO TO 11
      CALL RANDOM(NDR,.2,NRKE)
      NRD1=NTOT-NDR
      CALL RANDOM(NRD1,.1,NRKI)
      GO TO 12
   11 NRKE=0
      NRKI=0
   12 IF(NBESSS.LE.0) GO TO 10
C
C     *** BLUE ESCORT,INTD, AND RECCE KILLED ***
C     *** BY RED CAP ***
C
      CALL RANDOM(NBESSS,.15,NBEK)
      CALL RANDOM(NBRSS,.1,NBRK)
      CALL RANDOM(NBISS,.1,NBIK)
      GO TO 15
   10 NBEK=0
      CALL RANDOM(NBRSS,.15,NBRK)
      CALL RANDOM(NBISS,.2,NBIK)
   15 NBEK=NBEK+(NBES-NBESSS)
      NBRK=NBRK+(NBREC-NBRSS)
      NBIK=NBIK+(NBITD-NBISS)
      NBREF=.5*(NBREC-NBRK)
      NBIEF=.5*(NBITD-NBIK)
      BLOSS(2,9,5)=NBEK
      BLOSS(2,3,5)=NBRK
      IF(NBITD.LE.0) GO TO 22
      RATIO=FLOAT(BFORCE(1,4,5))/FLOAT(NBITD)
      JA=NBIK
C
C     *** UPDATE BLUE LOSS ARRAYS ***
C
      DO 25 I=1,JA
      RN=RANF(D)
      IF(RN.GT.RATIO) GO TO 26
```

```
      BLOSS(2,4,5)=BLOSS(2,4,5)+1
      GO TO 25
   26 RATIO1=FLOAT(BFORCE(1,4,5)+BFORCE(2,4,5))/FLOAT(NBITD)
      IF(RN.GT.RATIO1) GO TO 27
      BLOSS(2,4,5)=BLOSS(2,4,5)+1
      GO TO 25
   27 BLOSS(3,4,5)=BLOSS(3,4,5)+1
   25 CONTINUE
   22 NBD=AMINO(NRESSS,NBCAP)
      IF(NBD.LE.0) GO TO 21
      CALL RANDOM(NBD,.15,NBKE)
      NBD1=NBCAP-NBD
      CALL RANDOM(NBD1,.1,NBKI)
      GO TO 20
   21 NBKE=0
      NBKI=0
   20 IF(NRESSS.LE.0) GO TO 30
C
C    *** RED ESCORT,INTD, AND RECCE KILLED ***
C    *** BY BLUE CAP ***
C
      CALL RANDOM(NRESSS,.15,NREK)
      CALL RANDOM(NRRSS,.1,NRRK)
      CALL RANDOM(NRISS,.1,NRIK)
      GO TO 35
   30 NREK=0
      CALL RANDOM(NRRSS,.15,NRRK)
      CALL RANDOM(NRISS,.2,NRIK)
   35 NREK=NREK+(NRES-NRESSS)
      NRRK=NRRK+(NRREC-NRRSS)
      NRIK=NRIK+(NRITD-NRISS)
      NRREF=.5*(NRREC-NRRK)
      NRIEF=.5*(NRITD-NRIK)
      IF(NRITD.LE.0) GO TO 40
      RATIO=FLOAT(RFORCE(1,4,5))/FLOAT(NRITD)
      JB=NRIK
C
C    *** UPDATE RED LOSS ARRAYS ***
C
      DO 41 I=1,JB
      RN=RANF(D)
      IF(RN.GT.RATIO) GO TO 42
      RLOSS(1,4,5)=RLOSS(1,4,5)+1
      GO TO 41
   42 RATIO2=FLOAT(RFORCE(1,4,5)+RFORCE(2,4,5))/FLOAT(NRITD)
      IF(RN.GT.RATIO2) GO TO 43
      RLOSS(2,4,5)=RLOSS(2,4,5)+1
      GO TO 41
   43 RLOSS(3,4,5)=RLOSS(3,4,5)+1
```

```
 41   CONTINUE
 40   IF(NRES.LE.0) GO TO 50
      RATIO=FLOAT(RFORCE(1,9,5))/FLOAT(NRES)
      JC=NREK
      DO 51 I=1,JC
      RN=RANF(D)
      IF(RN.GT.RATIO) GO TO 52
      RLOSS(1,9,5)=RLOSS(1,9,5)+1
      GO TO 51
 52   RLOSS(3,9,5)=RLOSS(3,9,5)+1
 51   CONTINUE
 50   IF(NRREC.LE.0) GO TO 60
      RATIO=FLOAT(RFORCE(2,3,5))/FLOAT(NRREC)
      JD=NRRK
      DO 61 I=1,JD
      RN=RANF(D)
      IF(RN.GT.RATIO) GO TO 62
      RLOSS(2,3,5)=RLOSS(2,3,5)+1
      GO TO 61
 62   RLOSS(3,3,5)=RLOSS(3,3,5)+1
 61   CONTINUE
 60   IF(REP.LE.0) GO TO 598
      PRK=1.-EXP((-.2*ND)/REP)
      NEP=REP
C
C    *** COMPUTE RED CAP KILLED ***
C
      CALL RANDOM(NEP,PRK,NRCAPK)
      RATIO=FLOAT(NRCAP(1))/FLOAT(NTOT)
      JM=NRCAPK+NRKE+NRKI
      DO 500 I=1,JM
      RN=RANF(D)
      IF(RN.GT.RATIO) GO TO 501
      RLOSS(2,5,1)=RLOSS(2,5,1)+1
      GO TO 500
 501  RLOSS(3,5,1)=RLOSS(3,5,1)+1
 500  CONTINUE
      GO TO 590
 598  NRCAPK=0
 590  IF(RCASEP.LE.0) GO TO 622
      PD=1.-EXP(-BEP/RCASEP)
      NRD=RCASEP*PD
      PCASK=1.-EXP((-.25*BEP)/RCASEP)
      CALL RANDOM(NRD,PCASK,NRCASK)
      NRTOT=NRSS(1)+NRSS(2)+NRSS(3)
      RATIO=FLOAT(NRSS(1))/FLOAT(NRTOT)
      JN=NRCASK
      DO 600 I=1,JN
      RN=RANF(D)
```

```
      IF(RN.GT.RATIO) GO TO 601
      RLOSS(1,6,1)=RLOSS(1,6,1)+1
      GO TO 600
  601 RATIOM=FLOAT(NRSS(1)+NRSS(2))/FLOAT(NRTOT)
      IF(RN.GT.RATIOM) GO TO 602
      RLOSS(2,6,1)=RLOSS(2,6,1)+1
      GO TO 600
  602 RLOSS(3,6,1)=RLOSS(3,6,1)+1
  600 CONTINUE
      GO TO 624
  622 NRCASK=0
      NRD=0
  624 IF(BEP.LE.0) GO TO 623
      PBK=1.-EXP((-.2*NRD)/BEP)
      NEP=BEP
C
C     *** COMPUTE BLUE CAP KILLED ***
C
      CALL RANDOM(NEP,PBK,NBCAPK)
      BLOSS(2,5,1)=BLOSS(2,5,1)+(NBCAPK+NBKE+NBKI)
      GO TO 699
  623 NBCAPK=0
  699 DO 700 I=1,3
      NCAS(I)=NBCAS(I)-BLOSS(I,6,1)
      BCAS(I)=FLOAT(NCAS(I))*BFRAC(I)
      BSUM=BSUM+BCAS(I)
      JCAS(I)=NRCAS(I)-RLOSS(I,6,1)
      RCAS(I)=FLOAT(JCAS(I))*RFRAC(I)
      RSUM=RSUM+RCAS(I)
  700 CONTINUE
C
C     *** COMPUTE FORCE RATIO ***
C
      FRATIO=(RARMY(2)*RSLOW+RSUM)/(BARMY(2)*BSLOW+BSUM)
      VAR=SIN(1.5708*((FRATIO-X1)/(X2-X1)))
      VEL=VAR**(2.*X3)
C
C     *** COMPUTE DAILY FEBA MOVEMENT ***
C
      FEBA=VMAX*VEL
C
C     *** COMPUTE TOTAL FEBA MOVEMENT ***
C
      TFEBA=TFEBA+FEBA
      RCASU1=1.-EXP(-BSUM/RARMY(2))
      RCASU2=RARMY(2)*RCASU1
C
C     *** RED ARMY CASUALTIES ***
C
```

```
      RCASU3=2.75-(.25*FRATIO)
      RCASU4=RCASU3/100.
      RCASU5=RARMY(2)*RCASU4
      BCASU1=1.-EXP(-RSUM/BARMY(2))
      BCASU2=BARMY(2)*BCASU1
C
C    *** BLUE ARMY CASUALTIES ***
C
      BCASU3=1.5*FRATIO+.5
      BCASU4=BCASU3/100.
      BCASU5=BARMY(2)*BCASU4
      TRATIO=RARMY(2)/BARMY(2)
      BLOST=BCASU2+BCASU5
      RLOST=RCASU2+RCASU5
      BA=BARMY(2)-BLOST
      RA=RARMY(2)-RLOST
C
C    *** CURRENT FORCE RATIO ***
C
      CRATIO=RA/BA
      NBCR=NBCAP-NBCAPK
      NRCR=NTOT-NRCAPK
      NTBCAS=0
      NTRCAS=0
      DO 900 I=1,3
      NTBCAS=NTBCAS+NBCAS(I)
      NTRCAS=NTRCAS+NRCAS(I)
  900 CONTINUE
C
C    *** NO.OF EACH TYPE REMAINING ***
C
      NRTOT=NRSS(1)+NRSS(2)+NRSS(3)
      NBTOT=NBSS(1)+NBSS(2)+NBSS(3)
      NBKILL=NTBCAS-NBTOT+NBCASK
      NBREM=NTBCAS-NBKILL
      NRKILL=NTRCAS-NRTOT+NRCASK
      NRREM=NTRCAS-NRKILL
      NB1=NBES-NBEK
      NB2=NBREC-NBRK
      NB3=NBITD-NBIK
      NR1=NRES-NREK
      NR2=NRREC-NRRK
      NR3=NRITD-NRIK
C
C    *** PRINT OUT RESULTS ***
C
      WRITE(6,800)
      WRITE(6,801)
      WRITE(6,802)
```

```
      WRITE(6,803) DAY
      WRITE(6,801)
      WRITE(6,800)
      WRITE(6,804)
      WRITE(6,805) BARMY(2),RARMY(2)
      WRITE(6,806) BLOST,RLOST
      WRITE(6,807) BA,RA
      WRITE(6,808) TRATIO
      WRITE(6,809) CRATIO
      WRITE(6,815) FEBA
      WRITE(6,816) TFEBA
      WRITE(6,810)
      WRITE(6,49)
49    FORMAT(1H0,"ENTER 1 TO CONTINUE",/,/)
      READ*,LETSGO
      WRITE(6,811) NBCAP,NBCAPK,NBCR
      WRITE(6,812) NTBCAS,NBKILL,NBREM
      WRITE(6,817) NBES,NBEK,NB1
      WRITE(6,818) NBREC,NBRK,NB2
      WRITE(6,819) NBITD,NBIK,NB3
      IF(NBREF.GT.0) WRITE(6,823) RSLOW
      IF(NBREF.LE.0) WRITE(6,824)
      WRITE(6,49)
      READ*,LETSGO
      WRITE(6,813) NTOT,NRCAPK,NRCR
      WRITE(6,814) NTRCAS,NRKILL,NRREM
      WRITE(6,820) NRES,NREK,NR1
      WRITE(6,821) NRREC,NRRK,NR2
      WRITE(6,822) NRITD,NRIK,NR3
      IF(NRREF.GT.0) WRITE(6,823) BSLOW
      IF(NRREF.LE.0) WRITE(6,825)
800   FORMAT(23X,"***************************")
801   FORMAT(23X,"*",25X,"*")
802   FORMAT(23X,"*",2X,"GROUND BATTLE RESULTS",2X,"*")
803   FORMAT(23X,"*",10X,"DAY",I3,9X,"*")
804   FORMAT(1H0,35X,"BLUE ARMY",6X,"RED ARMY")
805   FORMAT(5X,"NO. OF DIVISIONS",14X,F6.2,9X,F6.2)
806   FORMAT(5X,"NO. DIVISIONS LOST",12X,F6.2,9X,F6.2)
807   FORMAT(5X,"NO. DIVISIONS REMAINING",7X,F6.2,9X,F6.2)
808   FORMAT(5X,"BEGINNING FORCE RATIO",3X,F6.2)
809   FORMAT(5X,"CURRENT FORCE RATIO",5X,F6.2)
810   FORMAT(17X,"A/C ALLOCATED",4X,"A/C LOST",4X,"A/C "
     1"REMAINING")
811   FORMAT(5X,"BLUE CAP",9X,I4,9X,I4,11X,I4)
812   FORMAT(5X,"BLUE CAS",9X,I4,9X,I4,11X,I4)
813   FORMAT(5X,"RED  CAP",9X,I4,9X,I4,11X,I4)
814   FORMAT(5X,"RED  CAS",9X,I4,9X,I4,11X,I4)
815   FORMAT(5X,"TODAY'S FEBA MOVEMENT",3X,F6.2," KILOMETERS")
816   FORMAT(5X,"TOTAL FEBA MOVEMENT",5X,F6.2," KILOMETERS")
```

```
817 FORMAT(5X,"BLUE ESCORT",6X,I4,9X,I4,11X,I4)
818 FORMAT(5X,"BLUE RECCE",7X,I4,9X,I4,11X,I4)
819 FORMAT(5X,"BLUE INTDXN",6X,I4,9X,I4,11X,I4)
820 FORMAT(5X,"RED  ESCORT",6X,I4,9X,I4,11X,I4)
821 FORMAT(5X,"RED  RECCE",7X,I4,9X,I4,11X,I4)
822 FORMAT(5X,"RED  INTDXN",6X,I4,9X,I4,11X,I4)
823 FORMAT(5X,"TARGET STATUS IS ",F6.3,/)
824 FORMAT(5X,"NO INFO ON RED ARMY STATUS",/)
825 FORMAT(5X,"NO INFO ON BLUE ARMY STATUS")
    RETURN
    END
C
C     *** UPDATES FORCE ARRAYS, BASE STATUS, ***
C     *** LOGISTICS, AND SLOWDOWN ***
C
      SUBROUTINE RECAP(NCK,FORCE,LOSS,SIDE,NIEF,PA,NIE,SLOWDN,
     1BUF,XITORY,TYPE,ARMY,BASE,NSORT)
      INTEGER FORCE,DAY
      COMMON/A/ RSAM(2,6),BSAM(2,6),RSTAT(3),BSTAT(3),DAY
      COMMON/FIN/ NBACLS,NRACLS
      DIMENSION NTS(3),PART(3),NSUM(3),MOST(2,2),SUM(3),NIE(3)
      DIMENSION BUF(3),XITORY(3),TYPE(3),ARMY(3)
      DIMENSION FORCE(3,9,5),LOSS(3,9,5),BASE(5),NSORT(3)
      DATA MOST/210,190,280,850/
      PART(1)=PART(2)=PART(3)=0.
      DO 10 I=1,3
      NS=0
      DO 15 M=2,9
      IF(FORCE(I,M,1).GT.0) NS=NS+FORCE(I,M,1)
   15 CONTINUE
      NTS(I)=NS
      IF(FORCE(I,1,1).GT.0) PART(I)=NTS(I)/FORCE(I,1,1)
   10 CONTINUE
      DO 20 J=2,4
      K=J-1
      NSUM(K)=(FORCE(1,1,J)*PART(1))+(FORCE(2,1,J)*PART(2))+
     1(FORCE(3,1,J)*PART(3))
   20 CONTINUE
C
C     *** BASE SPARES = ON HAND - USED + ***
C     *** RESUPPLY ***
C
      DO 40 I=1,3
      XITORY(I)=XITORY(I)-(1.1*NSUM(I))
      XITORY(I)=XITORY(I)+(MOST(NCK,1)*(.995**NIE(I)))
   40 CONTINUE
C
C     *** SAME FOR ARMY ***
C
```

```
      ARMY(3)=ARMY(3)-(100.*ARMY(2))+(MOST(NCK,2)*(.995**NIEF))
      SLOWDN=ARMY(3)/(100.*PA)
      SLOWDN=AMIN1(1.,SLOWDN)
      DO 100 I=1,3
      DO 110 M=2,9
      DO 120 K=2,5
      LOSS(I,M,1)=LOSS(I,M,1)+LOSS(I,M,K)
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
C
C     *** SUM LOSS ARRAYS ***
C
      DO 200 I=1,3
      DO 210 M=2,9
      LOSS(I,1,1)=LOSS(I,1,1)+LOSS(I,M,1)
  210 CONTINUE
  200 CONTINUE
      DO 300 I=1,3
      JN=LOSS(I,1,1)
      DO 320 J=1,JN
      IF(FORCE(I,1,1).LE.0) GO TO 300
      RATIO=FLOAT(FORCE(I,1,2))/FLOAT(FORCE(I,1,1))
      RN=RANF(D)
      IF(RN.GT.RATIO) GO TO 321
      LOSS(I,1,2)=LOSS(I,1,2)+1
      GO TO 320
  321 RATIO1=FLOAT(FORCE(I,1,2)+FORCE(I,1,3))/FLOAT(FORCE(I,1,1))
      IF(RN.GT.RATIO1) GO TO 322
      LOSS(I,1,3)=LOSS(I,1,3)+1
      GO TO 320
  322 LOSS(I,1,4)=LOSS(I,1,4)+1
  320 CONTINUE
  300 CONTINUE
      SUM(1)=FORCE(1,1,2)+FORCE(2,1,2)+FORCE(3,1,2)
      SUM(2)=FORCE(1,1,3)+FORCE(2,1,3)+FORCE(3,1,3)
      SUM(3)=FORCE(1,1,4)+FORCE(2,1,4)+FORCE(3,1,4)
C
C     *** CHECK FOR LOGISTICS SHORTAGES ***
C
      DO 500 I=1,3
      BUF(I)=XITORY(I)/(SUM(I)*1.1)
      IF(BUF(I).GE.1.) GO TO 310
      IF(BUF(I).LT.1.) NSORT(I)=SUM(I)*BUF(I)
      IF(NCK.EQ.1) BUF(I)=AMIN1(BUF(I),BSTAT(I))
      IF(NCK.EQ.2) BUF(I)=AMIN1(BUF(I),RSTAT(I))
      GO TO 500
  310 IF(NCK.EQ.1) BUF(I)=AMIN1(BUF(I),BSTAT(I))
      IF(NCK.EQ.2) BUF(I)=AMIN1(BUF(I),RSTAT(I))
```

```
      NSORT(I)=220.*BUF(I)
  500 CONTINUE
C
C    *** PRINT OUT SUMMARY ***
C
      WRITE(6,189)
      WRITE(6,190) SIDE
      WRITE(6,191)
      WRITE(6,192)
      WRITE(6,191)
      DO 900 I=1,3
      NREM=FORCE(I,1,1)-LOSS(I,1,1)
      IF(NCK.EQ.1) NBACLS=NBACLS+LOSS(I,1,1)
      IF(NCK.EQ.2) NRACLS=NRACLS+LOSS(I,1,1)
      WRITE(6,193) TYPE(I),FORCE(I,1,1),LOSS(I,1,1),NREM
  900 CONTINUE
      WRITE(6,189)
      WRITE(6,194) SIDE
      WRITE(6,191)
      WRITE(6,195)
      WRITE(6,191)
      DO 901 I=1,3
      K=I+1
      WRITE(6,196) BASE(K),XITORY(I),BUF(I),NSORT(I)
  901 CONTINUE
      IFAKE=0
      WRITE(6,196) BASE(5),ARMY(3),SLOWDN,IFAKE
      WRITE(6,189)
  189 FORMAT(1X,50(1HX))
  190 FORMAT(15X,A5," AIRCRAFT STATUS")
  191 FORMAT(1X,"---------------------------------------------")
  192 FORMAT(4X,"TYPE",6X,"STARTING",6X,"LOSSES",6X,"REMAINING")
  193 FORMAT(4X,A5,8X,I3,10X,I3,11X,I3)
  194 FORMAT(15X,A5," BASE STATUS")
  195 FORMAT(4X,"BASE",6X,"NO. SPARES",6X,"STATUS",6X,"MAX ACFT")
  196 FORMAT(4X,A5,7X,F8.3,7X,F6.3,8X,I4)
      DO 400 I=1,3
      DO 410 K=1,5
      FORCE(I,1,K)=FORCE(I,1,K)-LOSS(I,1,K)
  410 CONTINUE
  400 CONTINUE
C
C    *** UPDATE FORCE ARRAY ***
C
      DO 600 I=1,3
      DO 610 M=2,9
      DO 620 K=1,5
      IF(FORCE(I,M,K).GT.0) FORCE(I,M,K)=0
  620 CONTINUE
```

```
  610 CONTINUE
  600 CONTINUE
C
C    *** RE-INITIALIZE LOSS ARRAY ***
C
      DO 700 I=1,3
      DO 710 M=1,9
      DO 720 K=1,5
      LOSS(I,M,K)=0
  720 CONTINUE
  710 CONTINUE
  700 CONTINUE
C
C    *** UPDATE SAM ARRAYS ***
C
      BSAM(1,6)=1.
      RSAM(1,6)=1.
      BSAM(2,6)=1.
      RSAM(2,6)=1.
      RSAM(1,1)=RSAM(1,1)+.5
      RSAM(2,1)=RSAM(2,1)+.5
      BSAM(1,1)=BSAM(1,1)+.5
      BSAM(2,1)=BSAM(2,1)+.5
      ARMY(2)=PA
      IF(NCK.EQ.1) GO TO 800
      DO 810 I=1,3
      IF(DAY.GT.1) RSTAT(I)=RSTAT(I)+.07
      RSTAT(I)=AMIN1(1.,RSTAT(I))
C
C    *** RED A/C REINFORCEMENTS ***
C
      DO 815 K=2,4
      NADD=8
      IF(I.EQ.3) NADD=16
      FORCE(I,1,K)=FORCE(I,1,K)+NADD
  815 CONTINUE
  810 CONTINUE
      FORCE(1,1,1)=FORCE(1,1,1)+24
      FORCE(2,1,1)=FORCE(2,1,1)+24
      FORCE(3,1,1)=FORCE(3,1,1)+48
C
C    *** ADVANCE THE DAY ***
C
      DAY=DAY+1
      GO TO 850
  800 DO 820 I=1,3
      IF(DAY.GT.1) BSTAT(I)=BSTAT(I)+.07
      BSTAT(I)=AMIN1(1.,BSTAT(I))
C
```

```
C     *** BLUE A/C REINFORCEMENTS ***
C
      DO 825 K=2,4
      NADD=9
      IF(I.EQ.2) NADD=17
      FORCE(I,1,K)=FORCE(I,1,K)+NADD
  825 CONTINUE
  820 CONTINUE
      FORCE(1,1,1)=FORCE(1,1,1)+27
      FORCE(2,1,1)=FORCE(2,1,1)+51
      FORCE(3,1,1)=FORCE(3,1,1)+27
  850 RETURN
      END
```

Bibliography

1. Berg, Donald J. and Marcia E. Strickland, Catalog of War Gaming and Military Simulation Models, SAGA-180-77. Washington D.C., August 1977.

2. Berkovitz, Leonard D., and Melvin Dresher, Optimal Employment of Tactical Air Forces in Theater Air Tasks - II: A Game Theoretic Analysis, RM-2137. Santa Monica, California, March 28, 1958.

3. Berkovitz, Leonard D. and Melvin Dresher, A Game Theory Analysis of Tactical Air War P-1592. Santa Monica, California, January 13, 1959.

4. Bracken, Jerome, Two Optimal Sortie Allocation Models Volume I: Methodology and Sample Results, Institute for Defense Analysis, Arlington, Virginia, December 1973. (AD 771 779).

5. Brewer, Garry D. and Martin Shubik, The War Game A Critique of Military Problem Solving, Cambridge: Harvard University Press, 1979.

6. Cordesman, Anthony H., Developments in Theater Level War Games, Washington D.C., December 1974. (AD A039 960)

7. Dadant, P.M., Measures of Effectiveness and the Tally/Totem Methodology, P-5062. Santa Monica, California, July 1973.

8. Emerson, D.E., The New TAGS Theater Air-Ground Warfare Model, R-1576-PR. Santa Monica, California, September 1974.

9. Forrester, J.W., Industrial Dynamics, Cambridge: MIT Press, 1961.

10. Harris, Kathleen and Louis H. Wegner, Tactical Airpower in NATO Contingencies: A Joint Air-Battle/Ground-Battle Model TALLY/TOTEM, R-1194-PR. Santa Monica, California, May 1974.

11. Hoeber, Francis P., Case Studies in Military Applications of Modeling, General Research Corporation, McLean, Virginia, August 1979.

12. Huber, Reiner K., Lynn F. Jones, and Egil Reine (Ed.), Military Strategy and Tactics: Computer Modeling of Land War Probalmes, New Yrok: Plenum Press, 1975.

13. Naylor, Thomas H. and J.M. Firger, "Verification of Computer Simulation Models," Management Science, 14: B92-B101 (October 1967).

14. Nosenzo, L.V., Prototype of a Theater-Level Tactical Combat Model, Lulejian and Associates, Torrance, California, February 1974. (AD 920 793)

15. Nussbaum, Daniel A., An Application of the Multivariate Analysis System to TAGS, Army Concepts Analysis Agency, Bethesda, Maryland, December 1976. (AD A034 521)

16. Overholt, John L., Analysis Data Inputs and Sensitivity Tests in War Games, Center for Naval Analyses, Arlington, Virginia, March 1971. (AD 722 858)

17. Reichenbach, Hans, The Rise of Scientific Philosophy, Berkeley: University of California Press, 1951.

18. Shannon, Robert E., Systems Simulation the Art and Science, Englewood Cliff: Prentice Hall, Inc., 1975.

19. Sisks, C.P., L.A. Giamboni, and J.R. Lind, Analytic Formulation of a Theater Air-Ground Warfare System (1953 Techniques), RM-1338. Santa Monica, California, September 13, 1954.

20. Takachenko, P.N., Mathematical Models of Combat Operations, Army Foreign Science and Technology Center, Charlottesville, Virginia, April 1973. (AD 764 109)

21. Van Horn, Richard L., "Validation of Simulation Results," Management Science, 17: 247-257 (January 1971).

22. White, Frederick J., A Model for a Two-Sided Tactical Air Employment Simulation, Air University: Maxwell AFB, Alabama, May 1976.

Vita

John Matthew Foley was born in Laconia, New Hampshire, on October 19, 1949. He graduated from Laconia High School in 1967 and attended the University of New Hampshire from which he graduated in 1971 with a Bachelor of Arts degree in Mathematics. He entered the Air Force in 1971 and attended Undergraduate Navigator Training. Upon successful completion of navigator training in 1972, he was assigned as a combat crewmember on the C-130E cargo transport aircraft at Ching Chuan Kang Air Base, Taiwan; Clark Air Base, Philippines; and Yokota Air Base, Japan. He entered the Air Force Institute of Technology's Graduate Program in Strategic and Tactical Sciences in August 1978.

He is married to the former Huang Chu Chin of Taipei, Taiwan. They presently have two children: Alice and Sherril.

Permanent Address: 2398 SE Aneci St.
Port St. Lucie, FLA
33452

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GST/OS/80M-2	2. GOVT ACCESSION NO. AD-A083 911	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) STAG: A TWO PERSON SIMULATED TACTICAL AIR WAR GAME		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John M. Foley Capt		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology(AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE March 1980
		13. NUMBER OF PAGES
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)

APPROVED FOR PUBLIC RELEASE AFR 190-17.

JOSEPH P. HIPPS, Major, USAF
Director of Public Affairs

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

War Games
Theater-Level Models
Computer Simulation

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

The model developed in this study is a highly aggregated theater-level game comprised of interaction equations which utilize the allocation of aircraft to various missions on a daily basis to obtain the outcome of an offensive versus defensive systems engagement. The simulation which supports the model consists of an interactive air portion and a parametric ground portion. The theater of operations consists of two sides with their respective air and ground forces. While the model produces credible outcomes, the major objective is to reflect the effect of strategy and employment tactics on the outcome of the battle.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20. (cont.)

The model is designed to provide individuals an opportunity to plan and conduct an air war and to test various air employment concepts. Existing war games are quite large and contain so many factors that the main effects of a player's employment decisions are confounded by the interactive effects of the factors. The war game described in this report has a limited number of factors so that it is easier for players to determine the main effects of their strategies. Included in the appendices are a user's guide, analytical formulations, and a source listing of the program.