2019-03

STATE SPACE ANALYSIS OF THE GENERAL ANALYTIC MINEFIELD EVALUATION TOOL: IN PURSUIT OF THE ADVANCED UNDERSEA WEAPON SYSTEM

Flick, Brian C.

Monterey, CA; Naval Postgraduate School

https://hdl.handle.net/10945/62250

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

SUPPLEMENTAL TO THESIS
(1 OF 6: GAMETUG)

STATE SPACE ANALYSIS OF THE GENERAL ANALYTIC MINEFIELD EVALUATION TOOL: IN PURSUIT OF THE ADVANCED UNDERSEA WEAPON SYSTEM

by

Brian C. Flick

March 2019

Thesis Advisor:	Thomas A. Hamrick
Co-Advisor:	Paul J. Sanchez
Second Reader:	Steven E. Pilnick

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL SURFACE WARFARE CENTER – PANAMA CITY
PANAMA CITY, FLORIDA 32407-7001

General Analytical Minefield Evaluation Tool (GAMET) User's Guide
Version 3.3

DECEMBER 2013

BRIAN C. BELTON

USAGE: GAMET is a complex software tool designed for use by trained, experienced analysts. Incorrect data inputs, inappropriate model application or using the tool for something other than its intended purpose may result in unexpected or inaccurate results. It is recommended that users partner or consult with the GAMET team at NSWC-PCD when conducting studies or analysis to ensure accurate results.

DESTRUCTION NOTICE - For unclassified, limited documents, destroy by any method that will prevent disclosure or reconstruction of the document.

The General Analytical Minefield Evaluation Tool (GAMET)
User's Guide - Version 3.3

Brian C. Belton

Overview

The General Analytical Minefield Evaluation Tool (GAMET) is an object-oriented, event-driven simulation used in the evaluation of minefield effectiveness, system performance trade-offs and transitor vulnerability. In GAMET, a "mine" is synonymous with a "node". In utilizing the simulation, a scenario is created to describe a case to be studied. This case is evaluated over a set of runs by varying parameters and gathering statistics (Monte Carlo technique). During a run, transitors are sent through the minefield, one per transit, and results are recorded. There are no program limitations on the number of mines/mine types, sensors/sensor types, weapons/weapon types that can be used in a case. Similarly, there are no program limitations on the number of transitors, transitor types or traffic patterms which can be used in a run. GAMET models both standalone mines and cooperative networked fields of sensors, relays and remotes weapons. Mulit-dimensional table look-up implementation provides robust, flexible performance representation for sensors, weapons and communications. The Windows PC-based software includes a graphical user interface (GUI) which provides scenario editing capability and on-screen animation. Outputs, written to disk, include standard minefield measures of effectiveness (MOEs) and other statistics of interest. GAMET has the ability to run in either Analysis or Planning mode. In Analysis mode, the number of nodes in the field is user-defined and remains constant. In Planning mode, a target MOE value is user-defined and the number of mines/sensor nodes needed to achieve this MOE value is automatically determined by GAMET.

Simulation Input

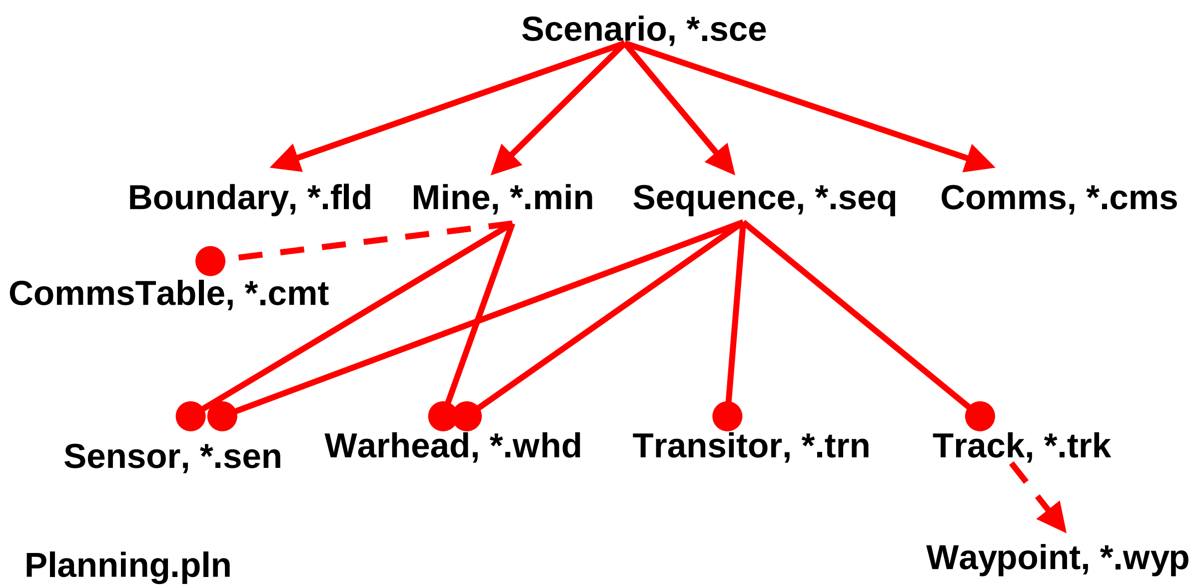
Data can be input to the program through data files, which are described below, and through the graphical user interface (GUI). It's recommended that all input files reside in GAMET's home directory. Input file types are distinguished by their three-letter extension. This allows the program to organize and automate input procedures. A sample set of input files is provided with the program and it is recommended that these files be copied and edited in order to minimize data format errors. The sample input files contain generic unclassified data. However, it's the user's responsibility to appropriately handle any classified input and output data used with the simulation. Comments and blank lines within these input files are for user readability only. However, while blank lines are skipped during file reads and can be implemented at the user's discretion, comments are expected by the program and must exist in the proper location. Input files necessary for the case should be closed prior to program execution, since some editors may not allow access to files that are open. This will cause the simulation to sit and wait until the necessary files are closed (released) before proceeding.

Input File	Extension	Description
Scenario	*.sce	Top level inputs. # of Runs and component filenames for case.

Batch	*.scb	Batch of multiple scenario files.
Boundary	*.fld	Coordinates defining minefield bounding rectangle.
Comms	*.cms	Comms/modem parameters
Comms Table	*.cmt	Table for probability of successful comms vs. range
Mine	*.min	Mine/node laydown; lines or random placement.
Sequence	*.seq	Sequence of transitor types per run.
Transitor	*.trn	Transitor type data; speed, keel depth, depth sigma.
Track	*.trk	Parameters for calculating transitor tracks through minefield.
Sensor	*.sen	Detection table for mine vs. transitor at a water depth.
Warhead	*.whd	Damage table for mine vs. transitor at a water depth.
Waypoint	*.wyp	Parameters for waypoint transit.
Planning	*.pln	Parameters for executing sim in planning mode

Input File Table

In order to run a GAMET simulation at least ten input files are required. Within GAMET, they are referenced in the following hierarchy. The scenario file designates one file of each of the types: Boundary, Comms, Mine, Sequence. These files form the second level of the hierarchy. By using only these file names on the second level, the Smart File Manager knows which files to read in for the simulation, saving the user from having to enter numerous filenames and reducing the chance for input errors. These files on the third level are named by using transitor type name, mine type name and water depth information contained in the Mine and Sequence files on the second hierarchical level. This approach provides the flexibility required for analysis of a wide variety of minefields, minefield depth profiles and transitor types. It also establishes a systematic way of naming the data sets required for the simulations. The number of input files required for a single simulation can be quite large. For instance, a simulation involving two mine types at three water depths with four transitor types will have 24 sensor, 24 warhead, 4 track files and 4 transitor files plus the comms table and the five files at the top two hierarchical levels for a total of 62 input files. Additionally, there may also be waypoint files if any of the transitors are using waypoint tracks. A planning file is also needed if executing the simulation in Planning mode. Naming details will be discussed for each file type.



Input File Hierarchy

Note: Some parameters may not be used in a simulation run, but the input files have standard templates which should be followed. This helps to eliminate errors and makes it easier to build new scenarios from existing file sets. The user should view these parameters as optional load-outs for simulation objects which can be easily adjusted (ie changing a mine group to 'Weaponized' will cause the weapon parameters to become active for that group, changing a field to 'Cooperative' will activate the comms parameters).

Scenario File

The scenario file is the top level input file. This file ends in the ".sce" extension and contains the number of runs in the case and the names of the component files which will be used to create the scenario. The first line in this file is a comment line to describe the scenario. The next line contains the number of runs for this particular case. The following four lines define the component files for the case; the boundary file, comms file, mine file and sequence file, respectively. In each data line, the data is preceded by a comment, which must be present.

Note: A large sampling size of runs (10,000 minimum recommended) should be used in order to ensure valid results. It is important to adhere to the format that exists in this file, as well as all other input files. Remember that for any data line, a comment preceding data cannot contain a space, since this is interpreted as the end of the comment string.

File Comment Line	
NumberOfRuns:	1000
BoundaryFile:	20x20.fld
CommsFile:	Comms5K.cms
MineFile:	Rising1lines.min
Sequence File:	4ShipA.seq

Sample Scenario File

Batch File

A scenario batch file may be used to run multiple scenario files. This file has the "*.scb" extension. The file starts with a comment line to describe the file. The next line defines how many scenario files are in the batch. Each of the following lines provides the name of a scenario file to be run (without any comments). These scenario files should reside in the same directory as their Batch file (ie no pathing within the batch file).

Note: In batch mode, GAMET will not prompt the user if output directories already exist and will be overwritten, in order to allow the batch of scenarios to run without the user present.

File Comment Line	
NumberOfFiles:	3
ShipSub&CoopRan.sce	
Sub&BottomRising.sce	
7SubA&Torpedo1.sce	

Sample Batch File

Boundary File

The name of the minefield boundary input file to be used in the case is found in the top-level scenario file. This file has the "*.fld" extension. The minefield boundary is a rectangle which must contain all mines in the scenario in order to ensure accurate results. The first line in this file is a comment line to describe the file. The next two lines describe the minefield's lower left and upper right corner (yd), respectively.

Note: The lower left minefield boundary should be set to (0,0) to provide a common reference point, minimizing input data errors and facilitating interchangeable data files. All mines must reside in the minefield box.

File Comment Line		
Minefield:xLo,yLo(yd):	0.0	0.0
Minefield:xHi,yHi(yd):	20000.0	20000.0

Sample Boundary File

Comms File

The name of the communication parameter input file to be used in the case is found in the top-level scenario file. This file has the ".cms" extension. This file describes basic communication parameters to be used in the scenario. The first line is the comment line to describe the file. The next lines define the node-to-node (hop) message processing time in seconds, the transmission retry delay (seconds), the maximum number of transmission attempts at low power (1 or more), the comms index for low power, the max transmission attempts at high power (0 or more) and the comms index for high power. The comms index values are used when interpolating from the Probability of Successful Communications tables (*.cmt). These tables are provided for each node type and provide the probability of successfully communicating given an index and a range. They are described in more detail later in this document.

Note: The comms tables can be filled with different data sets for probability of successful comms versus range and then the comms row index is used to designate which data sets (table rows) to use for that particular simulation study for low and high power. If the index value is between rows, interpolation will be done by the simulation.

File Comment Line	
NodePairProcessTime(sec):	10.0
TransmissionRetryDelay(sec):	5.0
MaxTransmissionAttemptsLowPower:	3
CommIndexLowPower:	3.0
MaxTransmissionAttemptsHiPower:	1
CommsIndexHiPower:	5.0

Sample Comms File

Mine File

The name of the mine/node laydown input file to be used in the case is found in the top-level scenario file. This file has a ".min" extension. This file describes the mine/node laydown for the scenario. Nodes are described in groups with certain characteristics (i.e. type, depth, …). The first line is a comment line to describe the file. The next line describes whether the field will have cooperative engagement capabilities, allowing a field of networked sensor, relay and weapon nodes to work together. In cooperative fields, non-weaponized nodes can pass detection messages to the nearest weaponized node through the communication network, allowing target prosecution by a remote weapon node. The next file line defines whether two remote detections of a target must be received at the same weapon node before a weapon will be launched (models a weapon node data fusion requirement). The next file line describes the criteria for which non-weaponized nodes will be replenished when a weapon node is replenished. The criteria can be either ADJACENT (non-weaponized nodes 1-hop from the replenished weapon node), LOWPOWER (non-weaponized nodes which have reached their low power replenish threshold), BOTH (adjacent and lowpower), or NONE. The next line defines how many mine/node groups there are in the field. A group is a set of nodes of the same type and characteristics which are placed in either a line or randomly within the minefield.

For each mine group, a set of parameters must then be provided in the file. They start with the node group ID, which is used in setting up node networks for cooperative fields. Only nodes with the same group ID can participate in a given network, allowing the creation of subnets. Next, the node type name is supplied and then a Boolean variable to describe whether this group's nodes are weaponized. The next line describes whether node groups will have "LINES", "RANDOM" of "LIST" placement. The next line sets the trigger type, either Probability of Fire (POF) or Ship Count (SC). The next two lines describe the Probability of Fire parameters, starting with the number of POF settings and then a parameter line defining the potential setting values (0-1). When the trigger type is POF, each node in the group will be assigned a POF randomly selected from the potential setting values. Identical values can be used to create ratios, ie Values: 0.5 0.5 0.8 will create a group where two thirds of the mines (on average) have a POF setting of 0.5. For each target detection, a random draw (0-1) is compared to the assigned POF to determine if a weapon fires. The next two lines describe ship count parameters, starting with the number of ship count settings and then a parameter line defining the setting values (integers). When the trigger type is SC, each node in the group will be assigned an initial ship count based on a random draw of these possible values. Identical values can be used to create ratios, ie Values: 1 3 3 will create a group where two thirds of the mines (on average) have a ship count setting of 3. The next line provides the inter-ship dead period, in minutes, which defines the minimum time allowed for a mine between actuations (ie decrementing the ship count). The following five lines include the power parameters; total power, power used per detection, power used per message transmission at low power, power used per message transmission at high power, power used to receive a message, and low power replenish threshold. The following parameter lines include the sensor&modem reliability (0-1) (this single random draw is applied to both the sensor and modem for cooperative fields), number of warheads, warhead replenish threshold, replenish time delay (min), warhead reliability (0-1), probability of sensor survival after warhead detonation (0-1) and the number of nodes in the group, respectively. The next two lines contain the node depth (ft) and water depth at the group's location (ft). If node placement is "LINES" then there are three more data entry lines for the group, which define the node line start point, end point, and position sigma (yd), respectively. Nodes are placed at equal spacing along the line and then this position is varied according to the input position sigma value in the x-y plane. In order to achieve a set of nodes normally distributed about a point, simply enter the same start and end point for the line. For groups with "RANDOM" node placement, the nodes are randomly placed within the minefield boundary area. For groups with "LIST" placement, the position sigma (yd) is provided next. This is followed by a comment line and then a list of the x,y locations for each node. This placement type is used when planned or measured location data is known for each node. If the position sigma is not set to zero, node position will then be varied according to the input position sigma value in the x-y plane.

Note: All nodes should reside in the minefield box in order to ensure accurate results. The water depth provided in this file is solely for selection of performance tables for the sensor and warhead. Type names should be checked for consistency and are case sensitive. Horizontal, (x,y) positions are always in yards while depth is always in feet. Multi-warhead nodes should only be used with the cooperative field setting, since standalone mines are destroyed upon weapon use. For cooperative fields, GAMET uses the NodeTypeName for automatically loading the Comms Table input files (*.cmt).

```
File Comment Line

NodesCooperative(TRUE,FALSE):                          TRUE
Require2DetectionsToLaunch(TRUE,FALSE)                 TRUE
ReplenishCriteria(ADJACENT,LOWPOWER,BOTH,NONE)         BOTH
NumberOfNodeGroups:                                    1

NodeGroupId                                            1
NodeTypeName:                                          TorpedoOnly
Weaponized(TRUE,FALSE):                                TRUE
NodePlacement(LINES,RANDOM,LIST):                      LINES
TriggerType(POF,SC):                                   POF
NumberOfPOFsettings                                    2
POFsettings(0-1):                                      0.8  0.6
NumberOfShipCountSettings:                             1
ShipCountSettings:                                     1
InterShipDeadPeriod(min):                              10.0
PowerTotal:                                            500.0
PowerDetect:                                           20.0
PowerTransmitLow:                                      10.0
PowerTransmitHi:                                       20.0
PowerReceive:                                          5.0
PowerReplenishThreshold                                50.0
Sensor&ModemReliability(0-1):                          0.9
NumberOfWarheads:                                      4
WarheadReplenishThreshold(-1=NoReplenish)              1
ReplenishTimeDelay(min)                                30.0
WarheadReliability(0-1):                               0.9
ProbOfSensorSurvival(0-1):                             0.95
NumberOfNodesInLine:                                   5
NodeDepth(ft):                                         580.0
WaterDepth(ft):                                        600.0
StartPt:x,y(yd):                                       5000.0    5000.0
EndPt:x,y(yd):                                         5000.0  15000.0
PositionSigma(yd):                                     100.0
```

Sample Mine File for One LINE Group

File Comment Line	
NodeGoupId	1
NodesCooperative(TRUE,FALSE):	FALSE
Require2DetectionsToLaunch(TRUE,FALSE)	FALSE
ReplenishCriteria(ADJACENT,LOWPOWER,BOTH,NONE)	NONE
NumberOfNodeGroups:	1
NodeTypeName:	Torpedo1
Weaponized(TRUE,FALSE):	TRUE
NodePlacement(LINES,RANDOM,LIST):	RANDOM
TriggerType(POF,SC):	SC
NumberOfPOFsettings	1
POFsettings(0-1):	0.75
NumberOfShipCountSettings:	3
ShipCountSettings:	1 3 3
InterShipDeadPeriod(min):	10.0
PowerTotal:	500.0
PowerDetect:	20.0
PowerTransmitLow:	10.0
PowerTransmitHi:	20.0
PowerReceive:	5.0
PowerReplenishThreshold	50.0
Sensor&ModemReliability(0-1):	0.9
NumberOfWarheads	1
WarheadReplenishThreshold(-1=NoReplenish)	-1
ReplenishTimeDelay(min)	30.0
WarheadReliability(0-1):	0.9
ProbOfSensorSurvival(0-1):	0.95
NumberOfNodesInGroup:	5
NodeDepth(ft):	580.0
WaterDepth(ft):	600.0

Sample Mine File for One RANDOM Group

File Comment Line	
NodesCooperative(TRUE,FALSE):	TRUE
Require2DetectionsToLaunch(TRUE,FALSE)	TRUE
ReplenishCriteria(ADJACENT,LOWPOWER,BOTH,NONE)	BOTH
NumberOfNodeGroups:	1
NodeGroupId	1
NodeTypeName:	TorpedoOnly
Weaponized(TRUE,FALSE):	TRUE

NodePlacement(LINES,RANDOM,LIST):	LIST
TriggerType(POF,SC):	POF
NumberOfPOFsettings	2
POFsettings(0-1):	0.8 0.6
NumberOfShipCountSettings:	1
ShipCountSettings:	1
InterShipDeadPeriod(min):	10.0
PowerTotal:	500.0
PowerDetect:	20.0
PowerTransmitLow:	10.0
PowerTransmitHi:	20.0
PowerReceive:	5.0
PowerReplenishThreshold	50.0
Sensor&ModemReliability(0-1):	0.9
NumberOfWarheads:	4
WarheadReplenishThreshold(-1=NoReplenish)	1
ReplenishTimeDelay(min)	30.0
WarheadReliability(0-1):	0.9
ProbOfSensorSurvival(0-1):	0.95
NumberOfNodesInLine:	5
NodeDepth(ft):	580.0
WaterDepth(ft):	600.0
PositionSigma(yd):	0.0
NodeList-XY(yds):	
2000.00 2000.00	
2000.00 4000.00	
2000.00 6000.00	
2000.00 8000.00	
2000.00 10000.00	

Sample Mine File for One LIST Group

Sequence File

The name of the transitor sequence input file to be used in the case is found in the top-level scenario file. This file has the ".seq" extension. The first line in the file is a comment line. The next line defines how many transitors are in a run. Each of the next lines contains a comment, transitor type, start delay and damage delay. The transitor types form the sequence of transitors in a run. GAMET uses this information to automatically load necessary input files, ie *.trn, *.trk. *.wyp. This prevents the user from having to coordinate the transitor types and enter the transitor files. The start delay indicates the number of minutes that transitor will wait before entering the battlespace if the previous transit ended without damage. The damage delay indicates the number of minutes that transitor will wait before entering the battlespace, if the previous transit ended with damage. A run ends either by damage or when the transitor reaches the exit point on the battlespace. Times for transit include the time in the battlespace, not just in the minefield, because areas inside the battlespace still represent a threat to the transitor, even if they lie outside the minefield, due to warhead range.

Note: GAMET automatically reads the input files based on the types given in the sequence file. Therefore, a transitor file, track file and waypoint file (if waypoints are being used) with the same name as the type given in the sequence file must be available to the simulation. i.e. If the sequence file lists a type named "Sub1" there must be files named "Sub1.trn", "Sub1.trk". Type names should be checked for consistency and are case sensitive.

File Comment Line			
NumberOfTransitors:	4		
TransitorType,Delay,DamageDelay(min):	Sub1	60.0	60.0
TransitorType,Delay,DamageDelay(min):	Sub1	5.0	10.0
TransitorType,Delay,DamageDelay(min):	Ship1	15.0	15.0
TransitorType,Delay,DamageDelay(min):	Ship1	5.0	10.0

Sample Sequence File

Transitor File

GAMET automatically reads the transitor files corresponding to the transitor types given in the sequence file. Therefore, for every transitor type listed in the sequence file, there must be a transitor file with that name available to the simulation. These files have the ".trn" extension. i.e. If the sequence file lists a transitor type named "Sub1" there must be a transitor file named "Sub1.trn". The first line in the file is a comment line. Three parameters must then be provided, one per line preceded by a comment. These are the transitor's speed (knots), it's keel depth (ft) and depth sigma (ft).

Note: Type names are case sensitive.

File Comment Line	
Speed(knots):	12.0
KeelDepth(ft):	150.0
DepthSigma(ft):	10.0

Sample Transitor File

WARNING: Ensure that the combination of keel depth and Sigma do not permit the target to run outside the range of depths specified in the .sen and/or.whd tables.

Track File

Track options for each transitor type are found in the track files. The program uses these files to determine the start and end point of each track. These files have the ".trk" extension and for every transitor type included in the scenario file, there must be a corresponding track file provided to the program. These files describe the entry of the transitor into the minefield by rectangle face. The faces are numbered 1 through 4, with the left face being 1 and then proceeding clockwise. The first line in the track file is a comment line. The next line contains the number of entry options the transitor can use to enter the minefield. The same entry face can be used multiple times. This provides the capability to have a variety of entries on the same side or to statistically “weight” sides relative to one another. For each face of entry, a set of parameters must then be provided to the program. The first line is the face number (1-4), the second line is the entry angle range, in degrees, with respect to the entry face (0-180), the next parameter line is the entry type (“bounded,channel,normal,random,waypt”). Depending on the type selected here, the user must do one of the following. If “random is selected, no more data is needed. If “normal” entry type is selected, two more lines must be provided for this face. They are the normal distribution center entry point (yd) on the face and the one-sigma value (yd) about this entry point. If “bounded” or “waypt” entry type is selected, two more lines must be provided defining the two points on the entry face between which the transitor will enter with uniform distribution. In the case where “channel” entry type is selected, three more lines must be provided. The first two are the points (one per line) that lie on the entry face, between which the first transitor will enter with uniform random distribution. The third input line is the sigma of the Gaussian random distribution that channelized transitors 2..n will have about this first transitor’s entry point.

Note: Care should be used to ensure that the selected entry point(s) do indeed lie on the face being described for normal, bounded, channel or waypt entry. Type names should be checked for consistency and are case sensitive. For “channel” and “waypt” traffic, only one entry option (face) is allowed per transitor type.

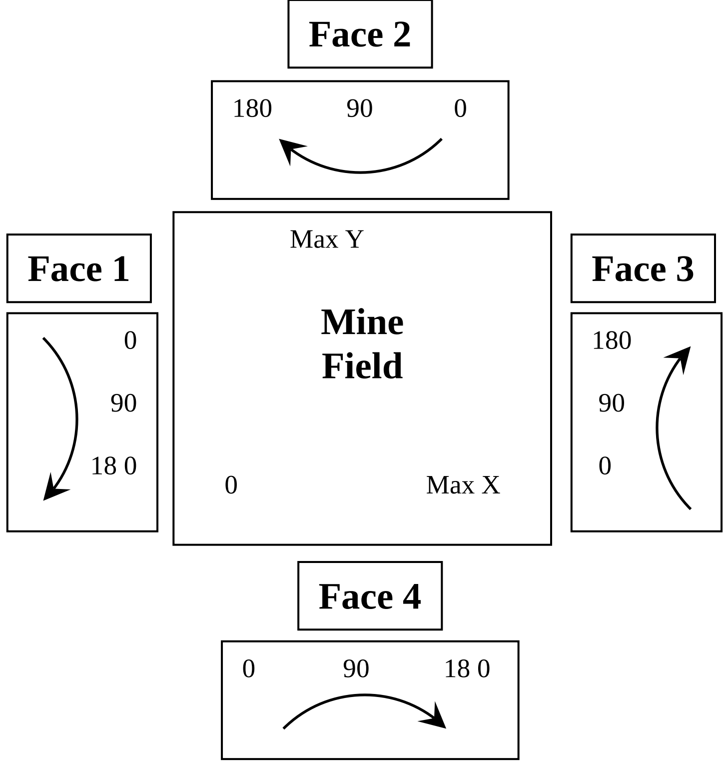
```
File Comment Line

NumberOfEntryOptions:                               2

EntryFaceNumber(1-4):                               1
EntryAngle:1,2(0-180degs):                          45.0    135.0
EntryType(bounded,channel,normal,random,waypt):     normal
EntryPoint:x,y(yd):                                 0.0     10000.0
EntrySigma(yd):                                     2000.0

EntryFaceNumber(1-4):                               1
EntryAngle:1,2(0-180 degs):                         90.0    90.0
EntryType(bounded,channel,normal,random,waypt):     random
```

Sample Track File



Waypoint File
A waypoint file must be included for each transitor type using a track of type "waypt". These waypoint files end with a ".wyp" extension and have names matching the transitor type name, ie "MCMShip.wyp". Within this file, the user provides a tabular listing of waypoints and whether the last waypoint is the end of the track or if the transitor is to continue. If the transitor is to continue, the user needs to specify whether: 1)The transitor is to go from the last waypoint to the first and repeat the sequence repetitively, or 2) The waypoints should all be run at random ad infinitum (ie loitering). Waypoints are shifted at each use by a normally distributed random navigation error in x and y with a specified sigma. A user-defined timeout must also be provided to prevent endless transits. Specific content within the file starts with the comment line. The next two lines contain the Boolean parameters for repeating the waypoint sequence until timeout and whether the waypoints should be run randomly. The next line defines the maximum leg length, in yards, that is allowed when random waypoints are being used. The next parameter lines define the amount of time for a transit before timeout occurs, in hours, and the position sigma error for the waypoints, in yards. The next parameter line defines the number of waypoints and is followed by a line for each waypoint, defining x and y, in yards.

Note: If the waypoints are not repeated then they must be run in order, ie if Repeat=FALSE then Random=FALSE.

```
File Comment Line

Repeat(TRUE/FALSE):                  TRUE
Random(TRUE/FALSE):                  FALSE
MaxLegLength(yd)(ForRandom=TRUE):    10000.0
Timeout(hr):                         240.0
```

PositionSigma(yd):	100.0
NumberOfWaypts:	3
Waypoint1:x,y(yd):	3000.0 3000.0
Waypoint2:x,y(yd):	5000.0 4000.0
Waypoint3:x,y(yd):	7000.0 8000.0

Sample Waypoint File

Communications Table File

The communications table file ends with a ".cmt" file extension. For each node type name, a *.cmt file with that same name must be provided if the minefield is cooperative. This file contains a probabilistic table for determining whether successful communications was achieved given a range and row index using 2-way interpolation. This table is used for comms at both low and high power by specifying an index (row) for each power level within the *.cms file. The *.cmt file contains no comments, in order to more easily accept data generated by separate programs. The first two values are integers defining the number of rows and columns of probabilities within the table. The next line contains ranges between the two communication points, in yards, starting at 0.0 and extending to the maximum communication range. The remaining lines in the table comprise the rest of the comms table. The first field in each of these lines is the comms row index, as specified in the Comms File (*.cms). The remaining fields in the line contain the probabilities corresponding to their row and column position.

Note: For cooperative minefields, there must be a *.cmt file for each node type. Also, within the table all probability rows should end with a 0.0 value. Nodes with zero communication range in the *.cmt table are considered to have no modem and will not be included in the network set up. Therefore, even nodes that act only as receivers must have a comms range > 0 in order to be included in the network. This file structure allows the user to mix different modem types, as well as standalone and networked mines, in the same field.

2 5					
	0.0	2999.9	3000.0	4999.9	5000.0
5.0	1.0	0.9	0.9	0.7	0.0
3.0	1.0	0.8	0.0	0.0	0.0

Sensor File

Sensor files end with a ".sen" file extension. For each transitor type versus mine type at a specific water depth, a sensor file containing a detection table must exist. GAMET automatically calculates the names of the sensor files from input data, saving the user from coordinating and entering numerous file names. This is done by concatenating the transitor type name (from the Sequence file), the mine type name and the water depth at the mine (from the Mine file), i.e. "Sub1&Rising1@60.sen". These files contain probabilistic tables used in determining transitor detection through 2-way linear interpolation. These files contain no comments, in order to more easily accept data generated by separate programs. The first two values are integers defining the number of rows and columns of probabilities within the table. The next line contains horizontal CPA ranges, in yards (from sensor to target), starting at 0.0 and extending to maximum detection range. The remaining lines in the table comprise the rest of the sensor file. The first field in each of these lines is the depth difference (delta z = mine depth – transitor depth), in feet, between the mine and transitor referenced from the mine's position (positive for transitor above mine). The remaining fields in the line contain the probabilities corresponding to their row and column position (CPA at delta z).

Note: Remember that CPA is expressed in yards and depth difference (delta z) is expressed in feet. Also, all probability rows must end with 0.0 as the last probability.

3	5				
	0.0	250.0	250.1	750.0	750.1
2250.0	1.0	1.0	0.8	0.8	0.0
0.0	1.0	1.0	0.8	0.8	0.0
-2250.0	1.0	1.0	0.8	0.8	0.0

Sample Sensor File

<u>Warhead File</u>

Warhead files end with a ".whd" file extension. For each transitor type versus mine type at a specific water depth, a warhead file containing a damage table must exist. GAMET automatically calculates the names of the warhead files from input data, saving the user from coordinating and entering numerous file names. This is done by concatenating the transitor type name, the mine type name and the water depth at the mine, i.e. "Sub1&Rising1@60.whd". These files contain probabilistic tables used in determining mission abort damage to the transitor through 2-way linear interpolation. These files contain no comments, in order to more easily accept data generated by separate programs. The first two values are integers defining the number of rows and columns of probabilities within the table. The next line contains horizontal CPA ranges, in yards (from warhead to target), starting at 0.0 and extending to maximum warhead range for this transitor type, speed, etc. The remaining lines in the table comprise the rest of the warhead file. The first field in each of these lines is the depth difference (delta z), in feet, between the mine and transitor referenced from the mine's position. The remaining fields in the line contain the probabilities corresponding to their row and column position (horizontal range at delta z).

Note: Remember that range is expressed in yards and depth difference (delta z) is expressed in feet. Also, all probability rows must end with 0.0 as the last probability. Currently, a warhead file must be supplied even for unweaponized mines (its performance can be set to all zeros).

3	4			
	0.0	500.0	750.0	750.1
2250.0	0.9	0.9	0.5	0.0
0.0	0.9	0.9	0.5	0.0

-2250.0	0.9	0.9	0.5	0.0

Sample Warhead File

Planning File

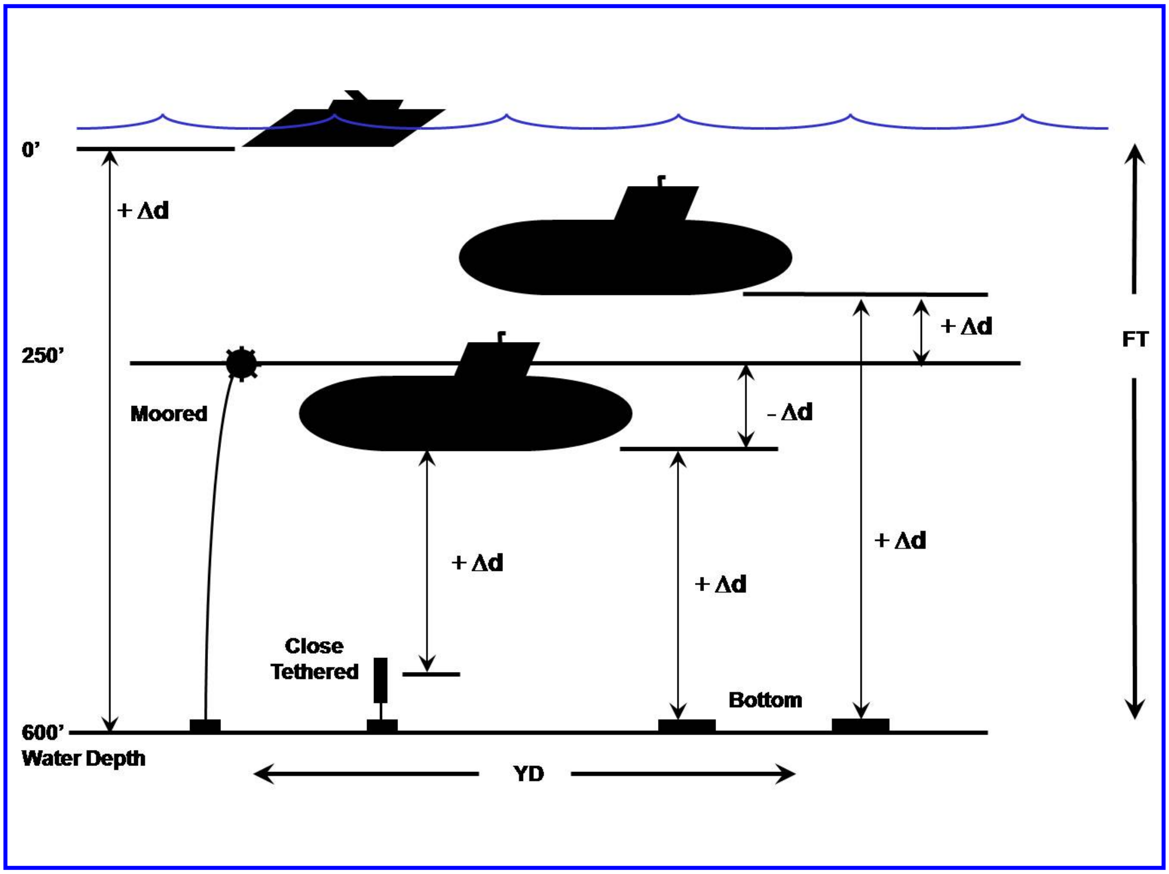
GAMET offers a Planning mode which allows the user to select a target MOE value and have the simulation increment the number of mines (or sensor nodes for cooperative fields) until the desired MOE is reached. In this mode, each mine group or sensor node group in the mine file is incremented by one each iteration until the desired MOE value is achieved. The StartIteration parameter determines the beginning number of nodes in these groups (1 per iteration). However, weapon node groups for cooperative fields, as well as all groups using LIST placement, don't vary their user-defined node count. The "Planning.pln" file holds the parameters used for Planning mode. The first line in the file is a comment header. The next line defines the maximum number of iterations to run in planning mode to try to achieve the target MOE value. The next line defines the StartIteration. The next line defines the target MOE type, either SIT (Simple Initial Threat) or EC (Expected Casualties). The last data line in the file defines the required MOE value the simulation must reach. These parameters are then presented through the GUI for editing by the user at run time. The final planning values which are used in the scenario will then be saved to the Planning.pln file which is written to the Inputs subdirectory for traceability.

Note: MOE names are case sensitive.

```
File Comment Line

MaxPlanningIterations:     100
StartIteration:            2
PlanningMOE(SIT,EC):       SIT
RequiredMOEvalue:          0.5
```

Sample Planning File



Range and Depth Measurements in GAMET

GUI Menu

The GUI provides many powerful features to the user. Pull-down menus at the top of the program window allow the user to edit scenarios and select animation options. The **Simulation** menu controls the start and stop of program execution. The **Mode** menu allows the user to choose between Analysis mode or Planning mode. The **Filename** menu allows the user to select filenames through a system file browser interface. There are four component files which are used to form a case; the minefield boundary, mine laydown, transitor sequence, and track file. By simply selecting a different component file, a completely new scenario can be generated. The user can view the selected case files through the "View All" option. The Filename menu also allows selection of a file editor (the default is "Notepad") through the "Editor" menu item

The **Edit** tab allows the user to edit the case files utilizing the selected file editor, as well as to edit key simulation parameters. The user may edit the run seed to regenerate a particular run number. Simply entering the desired run number (from a run beginning with the default seed of 1) as the seed will cause GAMET to begin the case at this run number. The GUI also allows the user to control the speed of program execution through the "Time Compression" option. This defines the real-time multiplier for execution and is helpful for slowing the animation to verify scenario inputs. To achieve maximum program execution speed, the program can be run without animation by using a -noimage command line option, i.e. "GAMET -noimage". A batch file GAMET-NoImage.bat is also included in the fileset which executes this option. In this mode, GAMET runs at maximum speed automatically.

The **Audio** tab allows the user to toggle the sound play feature "On" and "Off". The default is "Off". Currently sounds are played to denote a sensor's attempt to detect (ping.wav), a target detection (echo.wav) and a warhead detonation (detonate.wav).

The **Display** tab allows the user to display an overlay of the network field configuration.

The **Ouput** tab allows the user to toggle whether to output additional data files. The default is "Standard" where only the standard set of output files will be generated.

The **Help** tab provides GAMET Version and Developer information.

Menu	Option	Description
Simulation	Run	Begin sim execution after scenario selection.
	Exit	Halt the sim execution and exit.
Filename	Scenario	Select the top-level scenario file.
	Batch	Select a batch file of multiple scenario files.
	Comms	Select the communications file.
	Boundary	Select the minefield boundary file.
	Mine	Select the mine laydown file.
	Sequence	Select the transitor sequence file.
	View All	Display the number of runs and selected case files.
	Editor	Select a file editor (default=Notepad).
Edit	Scenario File	Edit a scenario file.
	Batch File	Edit a batch file.
	Boundary File	Edit the selected boundary file.
	Comms File	Edit the selected communications file.
	Mine File	Edit the selected mine file.
	Sequence File	Edit the selected sequence file.
	Track File	Edit one of the track files.
	Transitor File	Edit one of the transitor files.
	Sensor File	Edit one of the sensor files.
	Warhead File	Edit one of the warhead files.
	Waypoint File	Edit one of the waypoint files.
	Run Seed	Select the run seed (default=1). Allows rerun.
	Time Compression	Edit the animation speed (default=600 x real time).
	Number Of Runs	Edit the number of runs for the case.
Audio	On	Toggle the audio on.
	Off	Toggle the audio off (default).
Display	Network	Display an overlay of the network configuration

Output	Standard	Output standard result files (default)
	Expanded	Output all result files
Help	About GAMET	Show GAMET Version and Developer information

GUI Menu Options Table

Animated Display

An animated display capability is provided by GAMET to allow the user to visualize the scenario execution and verify input parameters. It also provides a way to capture or print a graphical sample of the minefield laydown and transitor track. These are useful for inclusion in briefings and as part of the permanent record of the simulation. Mines are represented by solid circles, whose color indicates the mine's state. Within the mine's circle is a smaller circle representing the warhead, with color indicating state. Sensors are represented by hollow circles centered at the mine's location. The displayed sensor radius is the maximum detection range as determined from the detection table, given transitor type. It does not ensure a detection, since detection is based on probability at transitor CPA, but merely allows filtering of objects to increase execution speed. If a detection is made, the sensor circle icon changes color for the remainder of the transit. For standalone mines, a warhead detonation is identified by an explosion icon on the screen, centered at its parent mine's location. For cooperative fields, a small explosion icon will appear at the weapon node for the weapon launch and a second, larger explosion icon will indicate transitor strike (if it occurs). If mission abort damage is delivered to the transitor, the transitor icon expands and then contracts, indicating the end of the transit. The program automatically selects the transitor icons based on the keel depth in the Transitor file, using a submarine icon for z >= 50.0 ft and ship icon for 0 ft <= z < 50.0 ft. There is also a plane icon for z < 0.0 ft (ie above the water surface) to represent low flying aircraft that are vulnerable to mines. For cooperative fields, white lines are used to indicate communication attempts (solid=success, dashed=failed) and a white receiver icon appears at the receiving node, to convey directionality. When a new message begins routing, icons related to previously completed messages are cleared from the screen.

At any time, the simulation can be paused with a single click on any menu item. A digital clock at the top of the window displays the simulation run time in hours:minutes:seconds. At the bottom of the display window, a status bar provides valuable information about menu options and program execution. The user can prevent the Windows Taskbar from covering this status bar by setting the Taskbar Properties to "Auto hide" or by maximizing the GAMET window. A description of the information provided in the status bar cells is given in the Windows Status Bar Table below.

Cell #	Description
1	Help statements describing menu options
2	Run number
3	Transit number
4	Transitor type
5	Program and mine vs. transitor status

Windows Status Bar Table

On-screen icons representing simulation objects can be selected during program execution by a mouse. Upon selection, the attributes corresponding to that object are displayed to the user. Selection of the minefield boundary box will toggle a zoom view of the minefield. On the display, icon colors are used to represent different object states. In the case of multiple warhead nodes, the icon represents the next warhead to be launched. Object icon descriptions are listed in the Object Icon Table below.

GAMET calculates a battlespace boundary box to ensure that all activities complete before the transitor leaves the battlespace. This ensures that accurate statistics are gathered. For cooperative fields, this battlespace buffer around the minefield can become quite large, depending on aspects of the communication network. Even with this boundary box, it's still possible to have active messages inside the field after the transit has ended (ie transitor destroyed), if the communication time parameters are set very high. This will force the simulation to wait a pre-calculated maximum possible field communication time. In this case, a warning message will be written to the DOS output window to alert the user because some statistics of interest may be affected.

Object	Color	Description
Mine	Red	Mine is armed.
	Light Red	Mine is armed and replenished.
	Green	Mine is unarmed.
	Dark Green	Mine is unarmed and replenished.
	Light Blue	Mine is destroyed by its warhead's detonation.
	White	Off/Out of power
Sensor	Purple	Sensor has not made a detection in this transit.
	Pink	Sensor has made a detection in this transit.
Warhead	Black	Warhead is present and on.
	Tan	Warhead is broken (determined unreliable at start of run).
	Gray	Warheads have been expended.
	Green	Warhead not present on mine (not weaponized).
Explosion	Fire	Warhead launch and detonation.
Transitor	Gray	Expand and shrink indicates mission abort damage received.
Track	Black	Track utilized by transitor.
Minefield	Black	Minefield bounding box.
Battlespace	Light Gray	Battlespace bounding box.
CommsLink	White-Dashed	Communications unsuccessful.
CommsLink	White-Solid	Communications successful.

Object Icon Table

<u>Error Checking</u>

GAMET performs limited error checking to aid the user in input data verification. If files required by the simulation are not available, an alert message identifying the missing file will be displayed on-screen. The animated display is also a great aid in identifying data errors and should be utilized as such. However, since the display is in 2D, data errors corresponding to depth cannot be visualized. Therefore, GAMET warns the user if any of the transitor depths are below the minimum water depth (as defined in the mine file). GAMET also warns the user if the output subdirectory already exists and asks the user for approval to overwrite output subdirectories.

Simulation Execution

At the start of each run, GAMET creates a node laydown by statistically perturbing the mine file inputs. These node positions remain constant throughout the run of transits. If the field is cooperative, GAMET then calculates the communication networks for the field, using Floyd's Shortest Path algorithm applied to time-path to a weapon node. Each network is built using nodes with the same NodeGroupId, allowing the creation of subnets within a field. The maximum comms range at low power, determined from the *.cmt file for that node type, is used to configure the network. If a node's modem is determined to be failed (by the sensor/modem reliability draw), that node will not be included in the network set up. Also, if a node has a communication range of zero specified in its *.cmt file, then it is considered to have no modem and is not included in the network set up (too allow a mix of standalone and networked mines).

For each transit, a track is calculated using the track file inputs. The transitor progresses through the minefield along its track, and the simulation pauses time at the closest point of approach (CPA) to all sensors where the transitor has entered the sensor's maximum detection range. This maximum detection range is determined from the table based on transitor type. The sensor then checks its detection table versus a random performance draw to determine if a detection has occurred. If a detection is made, the sensor notifies its parent mine. Given a detection, the mine checks the intership dead period timer and then decrements and checks the ship counter. If these criteria are satisfied, the mine then does one of several things. If the mines are not cooperative, then the mine tells its warhead to prosecute the transitor. A warhead detonation destroys the parent mine and sensor in this mode.

However, if the mines are cooperative, the mine (if it's a weapon node) attempts to prosecute the transitor itself. If it's not a weapon node, then it attempts to transmit a message down the communication path to the nearest (comms time-based) weapon node. It first uses low power comms to attempt to transmit the message one hop at a time. If a node-to-node communication hop is unsuccessful, the transmitting node will wait the Transmission Retry Delay time and then attempt to re-transmit, up to the user-specified number of Max Transmission Attemps Low Power. If that is unsuccessful, the node then tries to use high power comms to skip a hop and transmit the message to the following node. It tries this approach up to the Max Transmission Attempts Hi Power. If the message communication to the weapon node is successful, GAMET attempts to prosecute the target (ie launches warhead if armed and target is within max weapon range), and then uses the probability of sensor survival to determine if the weapon launch operation has destroyed the weapon node's sensor. If the user has specified Require2DetectionsToLaunch = TRUE, then for remote detections, two separate detection messages for a single transitor must be received by the weapon node, before a warhead will be launched. Once a warhead has launched, the warhead table is used to determine if mission abort damage is delivered to the transitor. If a weapon node reaches its Warhead Replenish Threshold, it triggers field replenishment. The weapon node, as well as any non-weaponized nodes which fit the user-specified Replenish Criteria, will be scheduled for replenishment, based on the Replenish Time Delay. If necessary, the replenishment activity will wait for a transit to complete in order to avoid field replenishment during a target's transit.

Power usage is tracked during the simulation. Each node begins with a specified number of power "units". Their remaining power level is then decremented for transitor detection attempts, and power to transmit and receive messages. These actions won't take place if there isn't sufficient power at the node. Once a node's power level reaches zero, it is turned off.

Note: Replenished nodes go through the same reliability draw process that happens at the start of a run for sensor/modem and warheads. The field network does not reconfigure after a replenishment. Therefore, if a "reliable" node is replenished and the replacement node's sensor/modem is statistically determined to be broken, nodes will continue to try to communicate to it in the network.

Simulation Output

Outputs of the GAMET simulation include standard minefield measures of effectiveness (MOEs). These outputs are saved in a set of standard files with the ".out" extension in a subdirectory of the scenario file location (and batch file location). This provides the option of keeping input/output data separate from the executable. This output subdirectory is given the name of the scenario file (without the extension). These files are described in the Output File Table below. In order to achieve statistically significant results, an ample number of runs must be run per case. Because GAMET is event-driven and runs rapidly, a minimum of 10,000 runs per case is recommended. At the end of the case, the threat profile and casualty density distributions are displayed graphically on-screen. If the simulation is being run in Planning mode, an output box will also appear that lists the Planning results.

Definitions and descriptions of some MOEs designated below are given in NWP 3-15. The StoppingCriteria.out file provides the Stopped Penetrator MOE probabilities as defined in:

> Timothy J. Horrigan, *Proceedings of the Sixteenth Technical Conference of the Naval Minefield Community (The Family of Underwater Weapons)* (U), NOLTR 73-73, June 1973, DTIC Accession No. C004899, (SECRET)

Timothy J. Horrigan, *Stopped Distributations, Sensitivity Settings and Ordnance Selection for Semiconfigured, Narrow Channels* (U), Horrigan Alalytics Report HA 75-1, July 1975, DTIC Accession No. C010157, (Secret)

Note: Since the output subdirectory is based on the scenario file name, running another case with the same scenario file name will cause the program to prompt the user to delete the old output files before rewriting the new output data. For traceability purposes, all input files used in the case are recorded in a subdirectory to the output directory named "Inputs". A "Scenario+timestamp.sce" file will also be created here which lists the actual scenario files and GAMET version used, in case the GUI was used to adjust input file selections for the scenario.

Output File	Contents
Threat.out	Logs the threat profile and expected casualties by transitor number.
Depeletion.out	Logs the effects of mine depletion within the minefield over a run of transitors. The first column denotes the number of mines removed from the field by detonation prior to the start of a particular transit, the second column provides the threat at this depletion level and the last column gives the number of transitors who transited at this depletion level.
Detonation.out	Logs the mine detonation statistics. Columns describe the number of lethal detonations, total number of detonations, number of runs and mines of each type in the case. Remaining columns describe detonation stats per run, including the max, min, expected and standard deviation of detonations and the corresponding mine type name.
Casualty.out	Logs the casualty density distribution. The first column indicates the exact number of casualties in a run and the second column indicates the probability of this occurrence over all the runs.
Scenario.out	This file is not an MOE. It logs the number of runs and the case files selected. It's identical in format to the *.sce files so that it can be rerun as the scenario input file to recreate the same case and for documentation of the run. These files are also recorded in the \Inputs subdirectory in the file "Scenario+timestamp.sce", which also records the GAMET version used in the case.
Threat_type.out	Logs the threat profile and expected casualties by transitor type. The transitor type name will be placed in the file name, ie Threat_SSN. The first column indicates the transitor number in the sequence for that type.
CasualtyDensity_type.out	Logs the casualty density distribution by transitor type. The first column indicates the transitor number in the sequence for that type. The remaining columns give the casualty density distribution.
CumCasualtyDensity_type.out	Logs the cumulative casualty density distribution by transitor

	type. The first column indicates the transitor number in the sequence for that type. The remaining columns give the cumulative casualty density distribution.
CommsStats.out	For each transitor, logs min, max and average time (in seconds) for successful end-to-end message communication (message creation to final destination), number of successes, total number of messages created and the average number of transmission attempts per hop for all messages (regardless of success). This file is only produced for cooperative field simulations.
NodeStatsAvg.out	Logs the location and statiststics for nodes, averaged over the set of runs. The first 2 columns are the x,y position of the nodes, with no error sigma. If Random mine entry is used, the positions of the nodes on the final run are recorded here. The next columns are power outages, power remaining at run end, warhead outages, warheads remaining at run end and number of times replenished. These stats are totaled over the set of runs for each node, then that number is divided by the number of runs.
NodeStatsByRun.out	Only output when in Test Mode. Logs the node location and statistics for each run. Columns are run#, x, y, power outages, power remaining, warhead outages, max # warheads, warheads remaining, and times replenished.
StoppingCriteria_type.out	Logs the probability of exactly J penetrators for an M casualty stopping criteria by transitor type. The first column indicates the number of penetrators of that type.
DistanceToCasualty.out	This file logs for each transitor type, the max, min, expected and standard deviation of distance to casualty, the number of casualties and transitor type, respectively. Distances are measured from the starting edge of the battlespace (threat area) to transit completion. Transits resulting in timeout (no casualty) are not included in the statistics.
TimeToCasualty.out	This file logs for each transitor type, the max, min, expected and standard deviation of time to casualty, the number of casualties and transitor type, respectively. All time within the battlespace (threat area) is counted in this statistic. Transits resulting in timeout (no casualty) are not included in the statistics.
TransitorFrequency.out	This file logs the time between transitors, as max, min, mean and standard deviation. Provides transitor frequency stats.
PlanningResults.out	This file is only generated in planning mode and logs the planning results, including total planning iterations, number of total nodes and weapon nodes, the actual MOE values and the user-required MOE value.

Output File Table