1-1-1991

GISMO: A Game For Intelligent Simulated Military Opponents

David R. Van Brackle

INSTITUTE FOR SIMULATION AND TRAINING

Contract Number N61339-89C-044
January 1991

GISMO: A Game for Intelligent Simulated Military Opponents

David Van Brackle • Donald Cross • Daniel Mullally • Eric Wampner

IST

Institute for Simulation and Training
12424 Research Parkway, Suite 300
Orlando FL 32826

University of Central Florida
Division of Sponsored Research

IST-TR-91-2

INSTITUTE FOR SIMULATION AND TRAINING

GISMO: A Game for Intelligent Simulated Military Opponents

David Van Brackle • Donald Cross • Daniel Mullally • Eric Wampner

Contract N61339-89C-044
January 1991

IST-TR-91-2

Institute for Simulation and Training
12424 Research Parkway, Suite 300
Orlando FL 32826

University of Central Florida
Division of Sponsored Research

* * * Table of Contents * * *

0 Purpose

This technical report is submitted as a deliverable item Number 2A, specified under Task 4, "Benchmark Development", of Workplan 4, dated 4 January, 1991 for DARPA contract N61339-89-C-0044, **INTELLIGENT SIMULATED FORCES: EVALUATION AND EXPLORATION OF COMPUTATIONAL AND HARDWARE STRATEGIES.** It discusses **GISMO**, the **G**ame for **I**ntelligent **S**imulated **M**ilitary **O**pponents, which acts as a testing platform for intelligent simulated force algorithms.

1 Introduction

GISMO is the **G**ame for **I**ntelligent **S**imulated **M**ilitary **O**pponents. It was created as a means of stimulating research into the fields of behavioral modelling of military forces and interactive distributed simulation. Competitors are invited to write programs to control a small force of tanks in a simple scenario. The programs communicate with a simulator via 2400-baud modems on toll-free telephone numbers, and compete interactively against one another.

GISMO is expected to have three distinct benefits: First, it will generate interest and stimulate research into intelligent simulated force algorithms. Second, it will generate and collect intelligent simulated force algorithms from a variety of sources. Third, it will serve as a testbed and simple benchmarking facility for intelligent simulated force algorithms.

2 Project Overview

The project consists of designing the competition, soliciting competitors, developing the hardware/software system, holding the competition and summarizing the results.

The competition consists of a simplified tank scenario. The tanks are abstractions of real tanks, and are capable of moving, turning their turrets, and firing a weapon. The battlefield is a 256x256 grid with rectangular terrain elements which may be forests, mountains or bodies of water. Each terrain element has an effect on the operability of tanks. The object of the scenario is to locate and destroy the opponent's blockhouse. Of course, the competing programs must prevent their own blockhouses from being found and destroyed. The terrain is not fixed, but may change from match to match.

The competitors in this competition are computer programs. The competing programs communicate with the simulator via 2400-baud modems on toll-free telephone numbers. Once connection and alignment are established, they each receive (over the modem) a description of the terrain for this particular scenario, and the location of their blockhouse. The battlefield is flat except for rectangular terrain elements of the type mentioned above. After sixty seconds, the competing programs must communicate to the simulator the initial placements of its tanks. From this point on, the competition proceeds in two-second time impulses until it ends. In each two-second time impulse, the simulator sends to each competing program a status message, and the competing programs send to the simulator orders for their tanks. The competition ends when one competing program destroys the opponent's blockhouse.

Any group or individual using any type of computing machinery will be allowed to compete. No human intervention is allowed once connection has been established between a competing system and the simulator.

The simulator is being developed on an HP Vectra 386 machine. Competitors will communicate with the simulator via 2400-baud modems on toll-free telephone numbers.

The competition will be held in October and November of 1991. This will give competitors almost a full year to develop their programs. The style of the competition (single elimination, double elimination, round-robin, swiss, etc.) will be determined by the number of competitors. The results will be announced at the 1991 IITSC, and the results will be released as a paper sometime later.

3 Hardware / Software Specifics

The main simulator will be implemented on an HP Vectra 386 machine in the C programming language. Two 2400-baud external modems will be attached to this machine, and will handle communication with competitors. The modems will be attached to telephone lines with toll-free numbers. Also connected to the HP machine will be an AT-class machine which will serve as a display device. Competitors may have access to this machine without affecting the main simulator. This machine will also have the capability of 'playing back' a completed match for display purposes.

There will also be a smaller version of the simulator. This version will be distributed to competitors. Several competing programs will also need to be written in order to fully test the system and to give competitors a 'sparring partner.' Lastly, a Terrain Editor must be developed. All software is being developed in Turbo C 2.0.

4 Workplan

The following is a description of the tasks necessary to complete GISMO. It should be noted that many of the tasks are already completed or are underway.

TASK 1: Design Competition

The rules of the competition should be simple and fair. Any group or individual using any kind of computing machinery should be allowed to participate. The rules should specify how conflicts are resolved and how an overall winner is chosen.

The scenario itself must be designed to be simple enough so that all pertinent information may be communicated quickly and easily, while still being complex enough to realistically simulate a tank battle.

TASK 2: Develop System

Once the scenario is designed, a simulator for it must be implemented. The hardware and system software must be procured and the simulator software written.

TASK 3: Solicit Competitors

Advertisements must be placed in major trade magazines for simulation and artificial intelligence. An information packet must be developed which can be sent to competitors upon expression of their interest. The competitors should be allowed to schedule practice bouts on our simulator in order to debug the communications aspects of their programs.

TASK 4: Hold Competition

The focal point of the project is the competition itself. A pairing scheme must be devised based on the number of competitors. The winner must be determined and announced, and awarded any prizes associated with the competition.

TASK 5: Summarize results

A paper should be written summarizing the results of the competition. It should include information about which teams did well, but more importantly why these programs did well. The body of this paper should be a cogent analysis of the algorithms submitted, how they fared and why they succeeded or failed.

5 Status

Tasks one through three have been completed. The game has been designed, and the simulator has been written. Seventy potential competitors have received descriptions of the game; ten have sent back completed Competitor Information Forms.

6 Appendices

Included in the appendices of this document are a copy of the rules of the competition and a copy of the description of the game scenario which were distributed to potential competitors.

Appendix A : Rules of the Competition

The following is a copy of the rules of the GISMO competition. It appears exactly as it was sent to potential competitors.

GISMO

GAME FOR INTELLIGENT SIMULATED MILITARY OPPONENTS

Rules

1. Any School, Company, Individual or Group of Individuals may enter, with the exception that students and employees affiliated with UCF or IST are ineligible.

2. Any hardware may be used.

3. The competition consists of writing a program to control a small force of tanks in a simple scenario. Programs communicate (via 2400-baud modems on toll-free telephone numbers) with a simple simulator at IST, and compete directly and interactively with each other.

4. A complete description of the game, simulator and communications parameters will be released after November 5, 1990. At any time after that date, a description of the game, simulator and communications parameters will be sent to competitors upon receipt at IST of an expression of interest by the competitors.

5. Competitors will have approximately one year to develop their programs. The competition will begin in October, 1991. Until that time, competitors will have access to the Game Simulator at IST for testing and practice bouts.

6. Competitors' programs must be completely self-reliant. Once connection to the IST Game Simulator has been established, no human intervention is allowed until the conflict is resolved.

7. A copy of all software and data used in the final program must be submitted in electronic form to IST before the contest finals begin. The software must be submitted in source code form. This includes all software and data used by the program, and any software which is not part of the final program but which was developed to be useful solely for developing the algorithms and data in the final program. This does not include general utilities developed in conjunction with the project which are not directly related to the nature of the game. The software may be proprietary, and IST guarantees that it will abide by any proprietary notices placed in the source code.

8. After programs and data have been submitted to IST, competitors may not modify them.

9. The style of the tournament (Round-Robin, Double Elimination, etc.) will be determined by the number of competitors. Initial pairings will be random.

10. Finals will be held at IST in late October and early November of 1991. Results will be announced and published.

11. A match will consist of six conflicts, with both competitors taking each side on each of three different terrains. Each conflict will be scored as follows:

Points:

4	Strategic Victory
3	Tactical Victory
2	Draw
1	Tactical Defeat
0	Strategic Defeat

12. If a competing program achieves victory conditions, it is awarded a Strategic Victory and its opponent is given a Strategic defeat.

13. If a competing program surrenders, it is given a Tactical Defeat and its opponent is awarded a Strategic Victory.

14. If a conflict or match is terminated due to a hardware failure at either competitors' site or at IST, the match will be rescheduled.

15. If a conflict is terminated due to a software failure in a competing program, the competitor experiencing the software failure is given a Strategic Defeat, and its opponent is awarded a Strategic Victory.

16. If it becomes apparent that neither competing program is capable of achieving victory conditions, the judges will terminate the conflict. Each competitor will be given a Tactical Victory, Tactical Defeat or a Draw depending on the strength of remaining forces.

17. In order for the rules to be properly enforced, it is desirable (but not always possible) for a representative of IST to be present with each competing program during a match. Arrangements will be made with each competitor on an individual basis at the appropriate time.

Appendix B : Game Description

The following is a copy of the description of the game to be used in the GISMO competition. It appears exactly as it was sent to potential competitors, including an errata sheet.

* * * Errata * * *

The Blockhouses section, paragraph 1, sentence 4 should be

> A Blockhouse is not damaged when a tank collides with it (although the tank is damaged.)

Communications section, paragraph 5, sentence 4 should be

> Tanks must be placed within a 25-grid radius of the player's blockhouse on Plain terrain. Tanks cannot be placed on any terrain other than Plain.

GISMO

GAME FOR INTELLIGENT SIMULATED MILITARY OPPONENTS

Game Description

Abstract:

GISMO is the Game for Intelligent Simulated Military Opponents. The goals of the project are to stimulate interest in Intelligent Simulated Forces, to collect and examine some ISF's, and to create a simple testbed in which ISF's may be compared to one another.

The goals are to be achieved by the creation of a game. Competitors are invited to write programs to 'Play the Game.' The competing programs communicate with a simple simulator via 2400-baud modems on toll-free telephone numbers. The simulator resolves conflicts and determines winners.

This document describes the parameters of the game which has been designed for the GISMO competition. It describes the overall scenario, the battlefield and terrain, the tanks, the methods of communication between competitors and the simulator, and various important algorithms used by the simulator (such as line-of-sight and checksum calculation.)

The Game:

The game is a simple battlefield simulation. In any conflict, each competitor will have a blockhouse, and will control seven tanks. The object of the game is to locate and destroy the enemy blockhouse.

Competing programs will communicate with the simulator via modem. First, they will receive a description of the terrain and the location of their blockhouse. They will then have one minute to formulate a plan. After the minute, they will be allowed to place their tanks. Tanks may be placed only on Plain terrain within a 25-grid radius of each player's blockhouse. Misplaced tanks will be placed 'near' their blockhouse by an arbitrary algorithm. Following tank placement, the competing programs will receive a status message from and give orders to their tanks once every two seconds until the game ends.

There are four ways for the game to end:

1) One competitor destroys the opponent's blockhouse
   Winner awarded Strategic Victory
   Loser given Strategic Defeat

2) One competitor destroys all opposing tanks
   Winner awarded Strategic Victory
   Loser given Strategic Defeat

3) One competitor surrenders
   Winner awarded Strategic Victory
   Loser given Tactical Defeat

4) Judges halt conflict (in case neither competitor has a chance of achieving victory conditions)
   Tactical Victory/Tactical Defeat or Draw assigned by Tiebreaker

The Tiebreaker is a numerical value calculated as follows: Score 10 points for each shell that hits the Blockhouse, and 1 point for each hit of damage sustained on each tank (including collisions & immobility due to water). The player with the **lower** score is awarded a Tactical Victory, and the opponent is given a Tactical Defeat. If the scores are equal, the game is ruled a Draw.

The Tanks:

On each 2-second time impulse, the competitors will receive a tank status report. This message will tell the position, heading and overall status of each tank. Competitors will also receive an object report describing the type, position and status of visible objects. (See the appendix for the format of the reports received by competitors from the simulator.)

Within two seconds of receiving these reports, competitors give orders for their tanks to the simulator. These orders have two parts: a Movement part consisting of a direction and speed, and a Weapons part consisting of a turret direction, whether or not to fire the weapon, and where to fire. Note that there is no "Scan" order, since this information is relayed once per impulse automatically. All of these actions would theoretically happen simultaneously, but to avoid confusion tanks obey orders in a fixed order: first firing, then changing speeds, then moving the tank, and finally turning the turret. All firing of all tanks during a time impulse occurs simultaneously. Damage is calculated after firing but before changing speeds.

First:	Fire
Then:	Change Speeds
Then:	Move
Then:	Turn Turret

Tank Movement:

The simulated tanks have fours speeds: Ahead Full, Ahead Half, Halted, Back Half. On any impulse, the tanks may accelerate or decelerate by only one speed (e.g. if moving Ahead Full, a tank could decelerate to Ahead Half, but could not Halt.) A tank cannot move at full speed while passing through Forest terrain, although it can move at full speed leaving and entering a forest. If a tank moves into a grid location with water, it immediately loses its mobility for the rest of the game, which is the equivalent of a hit of damage.

The tanks turn differently at different speeds (i.e. a tank moving Ahead Full cannot turn as sharply as a tank moving Ahead Half.) Thus, the result of a turn is based on the tank's speed as well as the direction it is currently facing. The tanks are limited to face in one of eight directions: N, NE, E, SE, S, SW, W, NW. The tanks' abilities to change direction at various speeds are summarized in tables to follow.

The Battlefield:

The battlefield will be a 256x256 cartesian grid, with (0,0) being the southwest corner and (255,0) being the southeast corner. At each point on the grid there will be an object or terrain. Objects can only be a Friendly Tank, a Friendly Blockhouse, and Enemy Tank, or an Enemy Blockhouse. Terrain is limited to: Plain, Forest, Water, Mountain. Tanks cannot move off the edge of the battlefield, but they will not be damaged for trying. The effect of terrain on tanks is as follows:

Tank Can:	Move Into	See Thru	Fire Thru
Plain	Yes	Yes	Yes
Forest	Yes*	No	No
Water	No**	Yes	Yes
Mountain	No	No	No

* Tanks cannot move at full speed through Forest.
** Tanks can move into water, but once there are immobile.

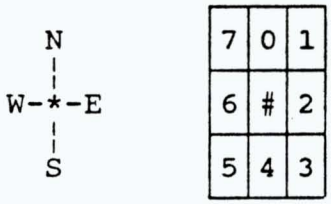
There is a special "non-object" called Smoke. It appears whenever a tank fires or is hit by a shell, and lasts for only one time impulse. It does not impede movement, vision or firing, and may appear at the same grid coordinates as another object. It simply serves as a visual confirmation of a tank firing or being hit. It can be seen over Water and Forest terrain, and other objects, but cannot be seen over Mountain terrain.

The Blockhouses:

Each competitor has one blockhouse. It cannot move and has no firepower. It does, however, report its status and what it can see just like a tank. It will be destroyed after being hit by three shells (at any distance). Blockhouses are not damaged when tanks collide with it (although the tank is damaged). When a blockhouse is destroyed, the game ends and the opponent is declared the winner.

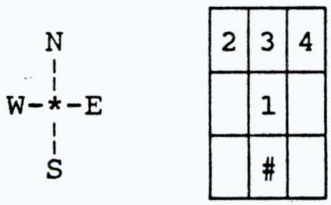
A moving tank can change direction by at most one compass point per time impulse. A Halted tank can change direction more quickly than a moving tank, turning up to two compass points per time impulse. A Halted tank facing N can turn to NE, E, NW or W. A Halted tank facing NE can turn to N, NW, E or SE. Of course, a Halted tank may also remain facing in the same direction.

A tank moving at half-speed moves one grid location per time impulse. The following table describes the Half-speed movement of a tank at #. In all cases, the tank's new heading is the same as its order.



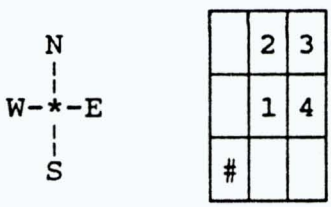
Heading	Speed	Order	Result
N	+1/2	NW	7
N	+1/2	N	0
N	+1/2	NE	1
N	-1/2	NW	3
N	-1/2	N	4
N	-1/2	NE	5
NE	+1/2	N	0
NE	+1/2	NE	1
NE	+1/2	E	2
NE	-1/2	N	4
NE	-1/2	NE	5
NE	-1/2	E	6

A tank moving at Full speed moves two grid locations per time impulse - one in its original direction, and one in its new direction (which may be the same as the original direction.)



Consider a tank at # facing N

Order	1st Move	2nd Move	Final Dir.
NW	N to 1	NW to 2	NW
N	N to 1	N to 3	N
NE	N to 1	NE to 4	NE



Consider a tank at # facing NE

Order	1st Move	2nd Move	Final Dir.
N	NE to 1	N to 2	N
NE	NE to 1	NE to 3	NE
E	NE to 1	E to 4	E

Weapons:

Each tank is equipped with a large gun attached to a rotating turret. Like the tank itself, the turret can be pointing in one of eight directions: N, S, E, W, NW, NE, SE or SW. The turret may rotate independently of the tank. In one time impulse the tank can be ordered to fire and then move its turret. The turret can turn up to 90 degrees relative to the tank per time impulse (this is the same turning rate as a halted tank.) The turret will turn with the tank (e.g. if the tank is facing N, the turret is facing W and the tank turns to face NW, the turret will now face SW.) For each direction, there is a scope of grid locations which can be hit (see the Vision/Firing Scope Diagram.) Each tank has a limit of 40 rounds of ammunition. Once a tank runs out of ammunition, it can no longer fire its gun. If the tank has not fired in the previous two time impulses, then the tank's ammunition loader is primed and can reload the gun quickly, and thus the tank may fire again on the next time impulse; if it has fired in either of the two previous time impulses and has fired in this time impulse, then it must take a time impulse to reload, so it cannot fire in the next time impulse. The gun has an effective range of 100 grid squares. A shot will either hit or miss its target (there are no "partial" hits), determined randomly. The probability of hitting a target is a function of distance, how fast the shooter is moving, how fast the target is moving, and from which direction the target is hit (see the appendices for a complete description of the To-Hit Probability Function.) Within a 50-grid radius, a hit is considered catastrophic, and thus destroys the target tank. From 50 to 100 grids, two hits are required to destroy the target tank (See **Damage** below.) Note that three shells are always required to destroy a blockhouse, even at close range. If an attempt is made to fire on a location which is not within the scope of the gun's current direction, the fire command is considered invalid. Thus, it is ignored, no ammunition is used, and the gun can be fired in the next time impulse without waiting for reloading. This is not the case for firing at a target which is out of range or obstructed.

When not firing, the tank's weapon crew is 'Tracking' a target. They can track targets only in the direction of the scope of their gun, and to which they have a clear line-of-sight. Whether a particular tank is firing its gun or tracking a target, the target location is given by its x- and y-coordinates in the tank's slot in the outgoing **Orders Packet.**

Line-of-Sight:

Many aspects of the game are affected by whether or not there is a clear line of sight between two points. The algorithm for determining grid locations lying on the line of sight between two points is outlined in the appendices. The following chart indicates which activities are inhibited by objects or terrain in the line of sight, and what type of objects or terrain inhibit them.

Action:	Blocked By:	Not Blocked By:
Seeing Tanks	Forest, Mountain, Tanks, Blockhouses	Plain, Water
Seeing a Blockhouse	Forest, Mountain, Blockhouses	Plain, Water, Tanks
Seeing Smoke	Mountain	Plain, Water, Forest, Tanks, Blockhouses

Note that a destroyed tank cannot see anything. Thus, this chart only applies to operational tanks.

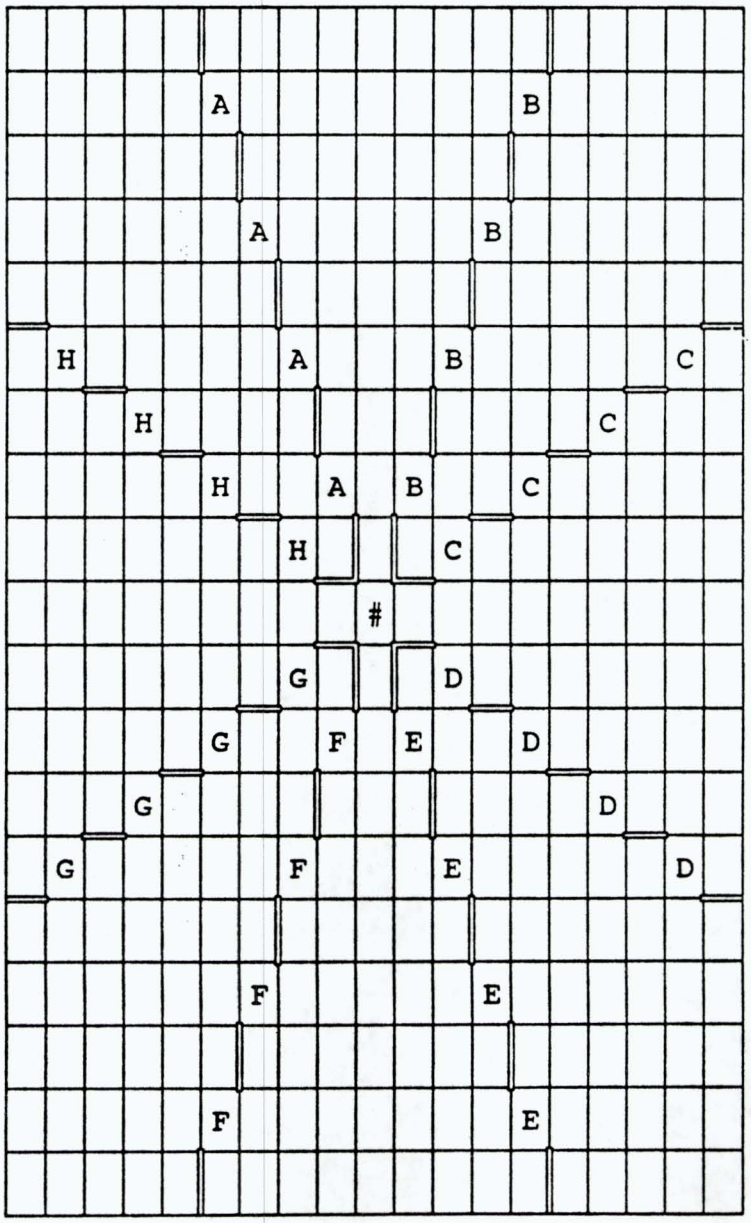
Tanks have limited angles of visibility. They can see only within one "wedge" of scope in the direction the tank is facing, and three "wedges" of scope centered in the direction the tank's turret is facing. The wedges referred to here are identical to those illustrated in the Vision/Firing Scope Diagram later in this document. For example, if a tank is facing North and its turret is facing South, the tank could see North, South, Southeast and Southwest.

It should particularly be noted that Line-Of-Sight is defined as grid locations **between** and **not including** two points. Thus, a tank on the edge of a forest has a clear Line-Of-Sight out of the forest.

A tank can fire at any tank or blockhouse if there is a clear line-of-sight between shooter and target. However, there is a special case when a tank may be within the line-of-sight of another tank but still cannot be seen. A tank on the edge of a forest may still not be seen even if another tank has a clear line-of-sight. If tank A is on the edge of a forest and tank B has a clear line-of-sight to tank A, then tank B still cannot see tank A if tank A is stationary, and can see tank A 50% of the time (chosen randomly) if tank A is moving. Note that Tank B can still fire at Tank A if there is a clear line-of-sight between them even if Tank A can't be seen.

Vision/Firing Scope Diagram:

The following diagram describes the scope of a turret aimed in a given direction. A letter in a square indicates that it can be hit by a gun facing in either of the border directions. For example, if the tank at # has its gun facing North, then it can hit any square labeled A or B, or anything in between. If its gun is facing Northeast, then it can hit anything labelled B or C, or anything in between. All the squares labelled B can be hit by the tank if its gun is facing North or Northeast.



The above diagram is summarized in the following table:

Let (X_F,Y_F) be the location of the Firing tank and (X_T,Y_T) be the location of the Target.

Let $Dx = X_T-X_F$, $Dy = Y_T-Y_F$. Then,

Gun Direction:	**Can Hit If:**
N	Dy >= -2Dx And Dy >= 2Dx
NE	Dy <= 2Dx And 2Dy >= Dx
E	2Dy <= Dx And 2Dy >= -Dx
SE	2Dy <= -Dx And Dy >= -2Dx
S	Dy <= -2Dx And Dy <= 2Dx
SW	2Dy <= Dx And Dy >= 2Dx
W	2Dy >= Dx And 2Dy <= -Dx
NW	2Dy >= -Dx And Dy <= -2Dx

Collision Resolution:

No two objects can occupy the same grid location (with the exception of the 'non-object' smoke). If a tank attempts to move into a grid location with another object or impassable terrain, or two tanks attempt to move into the same grid location, then a collision occurs.

If a tank is moving at half speed, then in any collision the following happens: the tank remains in its old location, but faces in the direction it attempted to move and permanently loses its mobility. This is the result if the tank attempts to move onto a mountain (forest and water are treated separately, however.) If a tank attempts to move into a grid location with a stationary tank, then both tanks lose their mobility for the rest of the game. If two (or more) tanks attempt to move into the same grid location, then all of the tanks collide and suffer mobility damage.

A full-speed move is equivalent to two half-speed moves. Therefore, collision resolution must occur in two phases: First, Half-speed and the first move of full-speed, and then the second move of full-speed (with all tanks not moving in a given phase considered stationary for that phase.)

The effect of one collision may cause another. Consider the following example:

		C	2
A	B	1	

If all three tanks are moving East and are ordered to continue moving straight, no collision occurs. Tank A moves into the space vacated by Tank B. However, if tank C is moving South, it will collide with B at 1. Since B doesn't move, it will then collide with A which is ordered into its location. Many tanks may be involved in such a collision.

Damage:

Tanks incur damage when they are hit by fire, or when they collide with other tanks, blockhouses or impassable terrain. They do not incur damage if they try to move off of the battlefield. Each tank can take two hits. The first hit causes the tank to lose all moving capability. The second hit causes the loss of weapons capability, rendering the tank destroyed. Once a tank is destroyed, it no longer reports objects within its line-of-sight, although it still reports its position. A shell fired from within 50-grid radius does 2 hits of damage to a tank, thus immediately destroying the target tank if successful. A shell fired from 50 to 100 grid squares does one hit of damage to a tank. A collision will destroy a tank's mobility, but not its weapon. Shells do the same amount of damage to a blockhouse regardless of the distance from which they were fired. Three shells are required to destroy a blockhouse under all circumstances.

It is possible for tanks (and blockhouses) to be hit by more than one shell during a single time impulse. When this happens, damage is computed in the usual manner, as though two successive hits had occurred. For example, suppose three of Player A's tanks simultaneously fire shells at one of Player B's undamaged tanks, and the distance between all of Player A's tanks and the target Player B tank is between 50 and 100 grids. The simulator determines (using the **"To Hit" Probability Function**) that two of the shells successfully hit the target, but one missed. The Player B tank would be destroyed, because the first shell would damage its mobility and the second one would damage its weapon.

Communications:

The simulator and the competitors communicate in units of information called packets. There are only a few kinds of packets, all of which are extensively documented in the appendices. Each packet begins with the string **GISMO** for alignment, and then each packet has a three-byte Protocol Block. The Protocol Block is a packet header which contains an ID of the packet, the length of the packet, and a checksum. Each packet sent by the simulator to competitors has a checksum which the competitors may use to verify the correctness of the packet. The checksum algorithm is given in the appendices. The competitors may or may not, at their option, include a checksum in the packets they send to the simulator. A Competitor never sends packets to the other competitor.

Before any communication can take place, the modems must establish connections. Once this has happened, the simulator will wait for the string **GISMO** to be sent by each competitor. This is for alignment purposes.

Once alignment has been established with both competitors, the simulator will send each competitor a **Startup/Terrain Packet.** This packet describes the battlefield terrain. The competitor should calculate the checksum of the **Startup/Terrain Packet** it has just received. (This is not mandatory, but it is strongly recommended.) If it matches the checksum in the Protocol Block, then the competitor should send a zero back in the Confirmation field of the **Terrain Acknowledge Packet.** If the checksums don't match, then the competitor should send back a non-zero value in the Confirmation field. After sending a **Startup/Terrain Packet**, the simulator will wait for a **Terrain Acknowledge Packet** from each competitor, looking for a success message. The simulator will try up to three times to send (and resend) the **Startup/Terrain Packet** to a competitor until it receives a success message in the subsequent **Terrain Acknowledge Packet.** If it fails three times, the simulation is aborted.

Once the simulator has received a correct **Terrain Acknowledge Packet** from both competitors, it will simultaneously send a **Blockhouse Location Packet** to each competitor. This tells the competitors the location on the battlefield of their blockhouse, and begins the one minute planning period.

Within one minute of sending a **Blockhouse Location Packet** to a competitor, the simulator will expect an **Orders Packet** from that competitor. The Firing field of this packet is ignored, and the i^{th} FireAtX and FireAtY fields are interpreted as the initial (x,y) coordinates of the i^{th} tank. The i^{th} Heading, Speed and Turret fields are interpreted as the initial heading, speed, and turret direction of the i^{th} tank. Tanks must be placed within a 25-grid radius of the player's blockhouse, on any terrain other than Plain. No tank can be placed at the same location as previously placed tanks or at the same location as the blockhouse. If a tank placement is faulty, the simulator will place the tank 'near' its blockhouse by an arbitrary algorithm.

From this point on, the simulation proceeds in two-second time impulses. The simulator will send a **Tank Report Packet** and an **Object Report Packet** to each competitor. The **Tank Report Packet** contains information on the location, speed, heading, turret direction and health of the competitor's tanks and blockhouse. It also contains information about the status of the game (e.g. if the game is over, if there are technical difficulties, etc.)

The **Object Report Packet** contains information about the objects that the competitor's tanks can see, what kinds of objects they are and how they are moving, and which tanks can and cannot see them. Note that a destroyed tank will not report objects in its line of sight, but it will still appear in the **Tank Report Packet**.

Within two seconds of having sent both report packets to a competitor, the simulator expects either an **Orders Packet** or a **Surrender Packet** from the competitor. Then, after resolving the move, a **Tank Report Packet** and an **Object Report Packet** are again sent to the competitor, in that order. This sequence continues at two-second intervals until the game is ended. If the game ends, this fact and how it ended are contained in the **Tank Report Packet**.

During the game, there are several ways that communications can become disrupted. In most situations when the simulator detects a bad stream of data from one of the competitors, it will simply ignore the data. Bad streams of data include missing **GISMO** alignment strings, incorrect checksums in the packets, or invalid orders in an otherwise correctly-formed packet.

When the simulator finds no **GISMO** string where it expects the beginning of a packet from the competitor, it will go into a loop waiting for **GISMO**. The simulator will count up the number of bytes that are received before finding the beginning of a **GISMO** string. If this number reaches 100 for a given packet, the game will be aborted due to "Technical Difficulties". Note that this counter is reset to zero every time **GISMO** is successfully found. The game will also be aborted if the number of instances of finding spurious data before a **GISMO** string reaches 3.

When the simulator receives a packet which has a nonzero value in the checksum field of the protocol block, it will compute the checksum of the data after the protocol block in the packet to make sure that it matches the transmitted checksum. See the appendices for the algorithm to compute checksums of packets. Note that this algorithm never produces zero as the value of a checksum, so that zero can be used as a flag for "ignore checksum". Each time a packet is received with a mismatch between the transmitted checksum and the computed checksum, a counter is incremented. If this counter reaches 3, the game is aborted due to "Technical Difficulties".

* * * Appendices * * *

"To Hit" Probability Function:

If the distance D between the shooter and the target is greater than 100 grid squares, then the probability of hitting is automatically zero. If not, it is calculated in the following way: Let (X_F,Y_F) be the location of the Firing tank, and (X_T,Y_T) be the location of the Target. Let $Dx = X_T-X_F$, $Dy = Y_T-Y_F$, $D^2=Dx^2 + Dy^2$ (D^2 is the square of the distance between the source and the target.) Then the base probability (expressed as a percentage) of hitting is

$$P = \text{Round}(100 * \exp(-3\text{x}10^{-4} * D^2) - 5)$$

This base probability is then adjusted by adding the following modifiers:

Speed of Tank:	Modifier:	Speed of Target:	Modifier:
-0.5	-3	-0.5	-5
0.0	0	0.0	0
0.5	-3	0.5	-5
1.0	-5	1.0	-10

If tank changed speed this turn:	-5
If tank changed direction this turn:	-5
If target changed speed this turn:	-5
If target changed direction this turn:	-5

Direction of Impact: (only applies if target is a tank.) Tanks are more heavily armored in front. Thus, the direction from which a shell hits affects the probability that the hit will do significant damage. For example, a head-on shot incurs a -5 point penalty, while a rear-end shot gets a +5 point bonus.) Let # be the target tank, and let 0 represent the direction the tank is facing:

```
7 0 1          6 7 0
 \|/            \|/
6-#-2          5-#-1
 /|\            /|\
5 4 3          4 3 2
```

Shot from:	0	1	2	3	4	5	6	7
Modifier:	-10	-5	0	+5	+10	+5	0	-5

If target was tracked **or** fired at last turn: +10

If the adjusted P > 95, then P = 95. (The probability of hitting can **never** be greater than 95%)

Chart of Hit Probability as a Function of Distance:

The following chart shows the base probability P of hitting a target (expressed in percent) as a function of distance D (expressed in grids). This chart is merely a table of values computed using the formula for P provided on the previous page. Note that this chart does not include the effects of any of the probability modifiers.

D	P	D	P	D	P	D	P
1	95	26	77	51	41	76	13
2	95	27	75	52	39	77	12
3	95	28	74	53	38	78	11
4	95	29	73	54	37	79	10
5	94	30	71	55	35	80	10
6	94	31	70	56	34	81	9
7	94	32	69	57	33	82	8
8	93	33	67	58	31	83	8
9	93	34	66	59	30	84	7
10	92	35	64	60	29	85	6
11	91	36	63	61	28	86	6
12	91	37	61	62	27	87	5
13	90	38	60	63	25	88	5
14	89	39	58	64	24	89	4
15	88	40	57	65	23	90	4
16	88	41	55	66	22	91	3
17	87	42	54	67	21	92	3
18	86	43	52	68	20	93	2
19	85	44	51	69	19	94	2
20	84	45	49	70	18	95	2
21	83	46	48	71	17	96	1
22	81	47	47	72	16	97	1
23	80	48	45	73	15	98	1
24	79	49	44	74	14	99	0
25	78	50	42	75	13	100	0

Line-Of-Sight Algorithm:

The following algorithm is used to determine the line-of-sight between points (X_0,Y_0) and (X_1,Y_1).

Let $Dx = X_1 - X_0$, $Dy = Y_1 - Y_0$.

The algorithm varies if $|Dx| >= |Dy|$ or not, if $Dx >= 0$ or not, and if $Dy >= 0$ or not. Thus, there are eight variations. Only the variation in which $|Dx| >= |Dy|$, $Dx >= 0$, $Dy >= 0$ will be given.

```
x = 1
y = 0
Blocked = False
While x<Dx AND NOT Blocked
     If (y+1)*Dx <= x*Dy + Dx DIV 2 Then
          y = y + 1
     End If
     If Battlefield[X0+x,Y0+y] is a Blocker Then
          Blocked = True
     End If
     x = x + 1
End While
```

If $|Dx| < |Dy|$ then the roles of x and y in the above algorithm are reversed. If $Dx < 0$ (or $Dy < 0$) then x (or y) is decremented rather than incremented, and appropriate changes must be made to the While and If conditions. It should be noted that all values are integers, and that Dx DIV 2 represents an integer division.

Checksum Algorithm:

Checksums are calculated for the data part of a packet only. Protocol Blocks are **not** figured into the checksum. Checksums are calculated with the following algorithm:

```
Byte Checksum = 0
For each data byte of message, in order, Do
     Flip All 8 Bits of Checksum (Bitwise NOT)
     Rotate Checksum Left 1 Bit (MSB becomes LSB)
     XOR Checksum with Next Data Byte
End For
If Checksum == 0 Then Checksum = 255
```

Note that this checksum algorithm is sensitive to the order of the data bytes. Thus, the checksum must be calculated in order from the first data byte of a packet to the last. Also, this algorithm will never produce a zero value for the checksum. This is because the GISMO protocol uses zero a flag to signify that the checksum field is to be ignored.

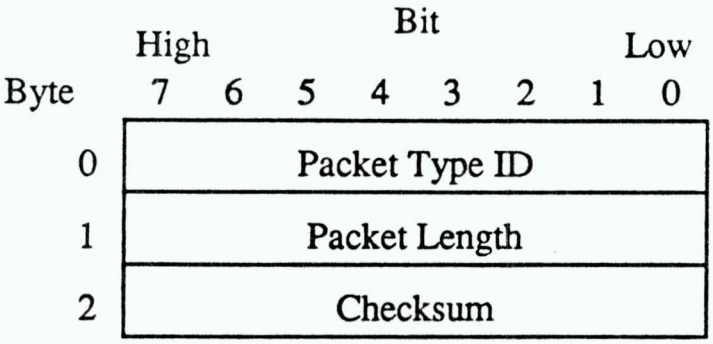
Communications Parameters:

GISMO's communications will occur along 2400-baud modems. Speeds greater than or less than 2400 baud will not be supported. Competing programs should use No Parity, 8 Data Bits, 1 Stop Bit.

Packets:

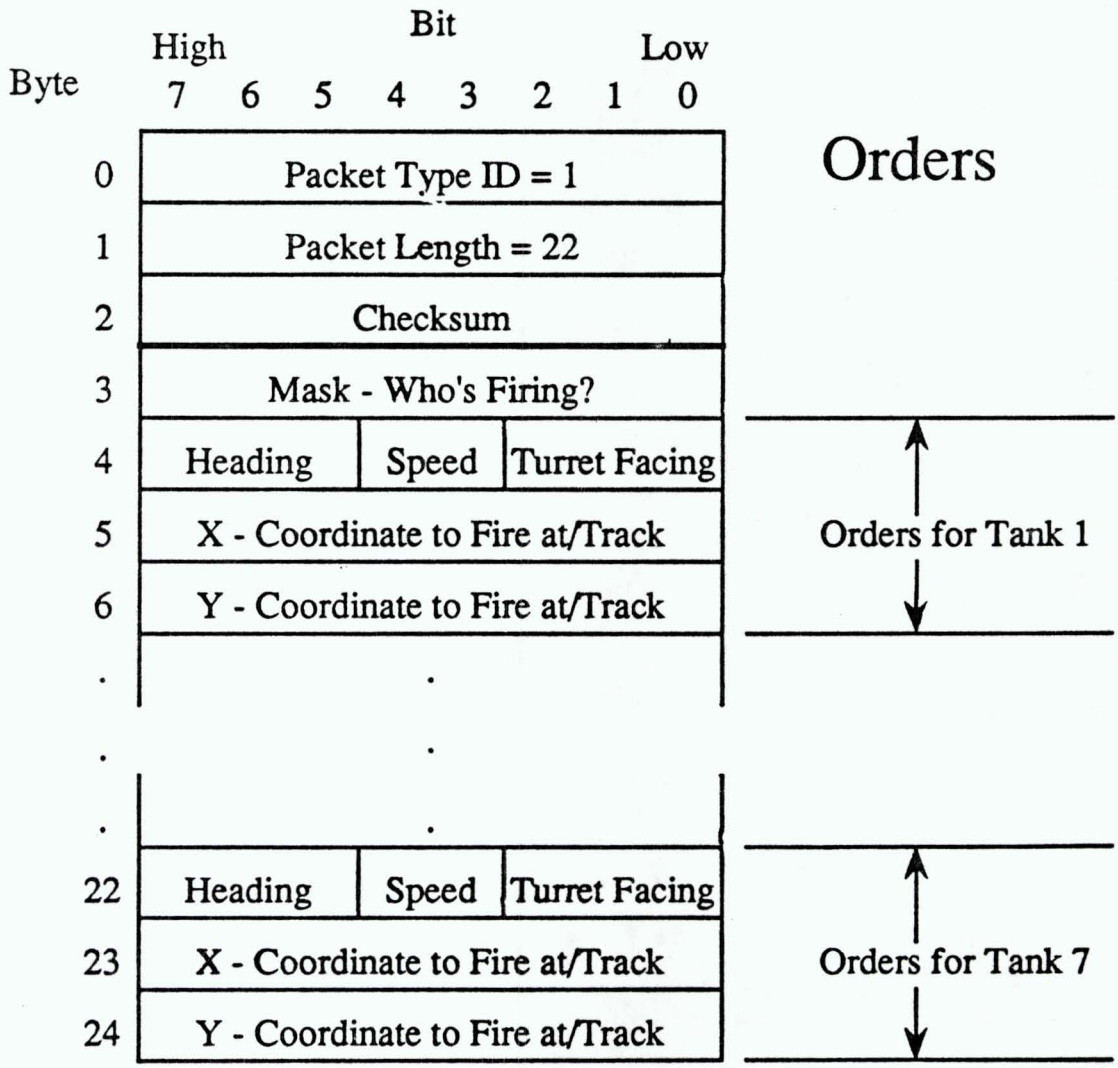
A description of the packets follows. Note that for alignment each packet must be preceded by the string **GISMO**. Part of the pertinent information is summarized in the table below - note that the Length is in bytes, and does not include the 5-byte **GISMO** or the 3-byte Protocol Block.

Packet:	Length:	Packet Type ID:
Orders	22	1
Tank Report	29	2
Object Report N = number of objects (max 24)	1 + 5*N	3
Startup/ Terrain N = number of terrain elements (max 255)	1 + 5*N	4
Blockhouse Location	2	5
Terrain Acknowledge	1	6
Surrender	9	7



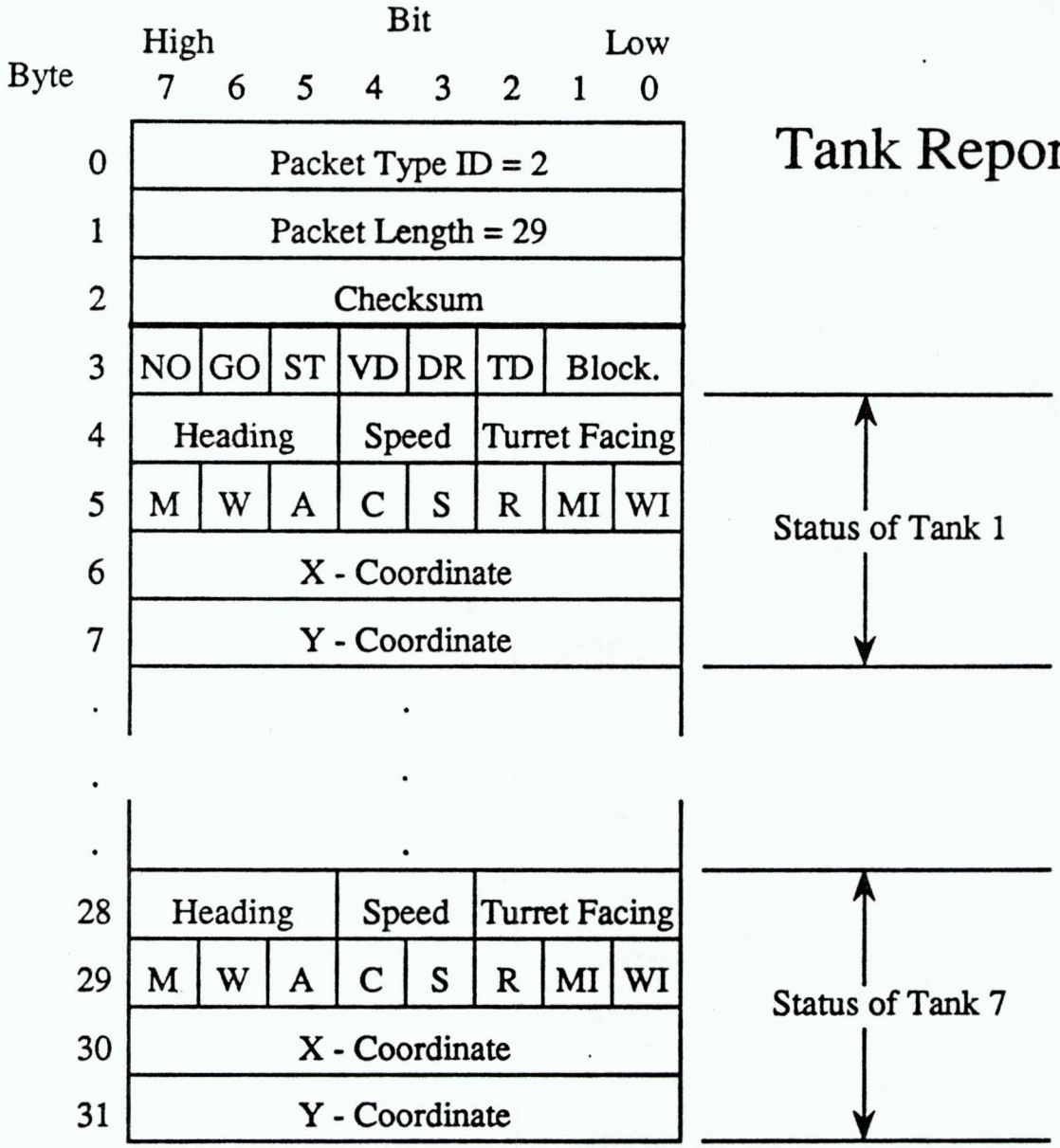
Protocol Block

Packet Length field does not include length of the **GISMO** or of the protocol block itself.



Orders

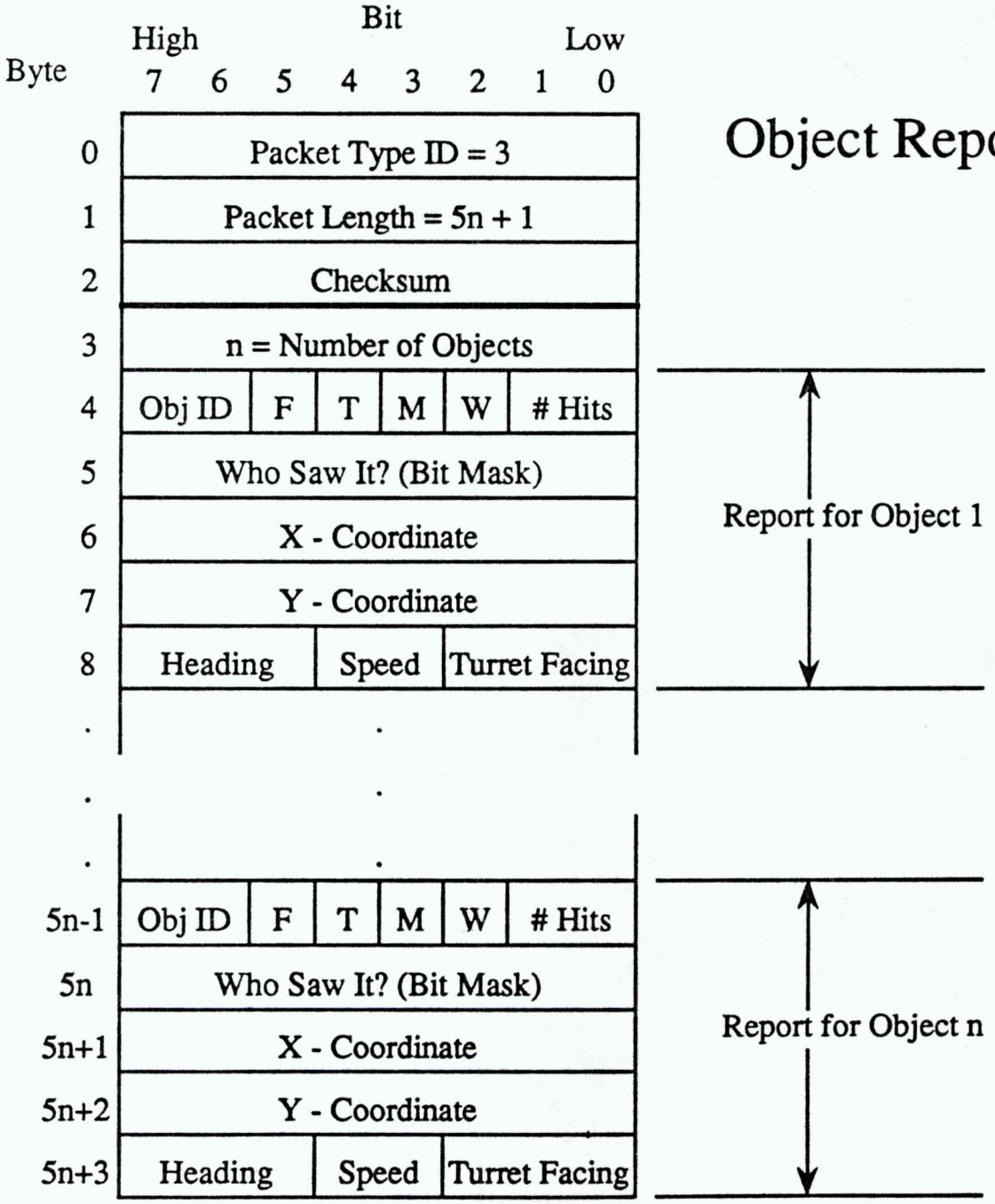
Tank Report



NO : 1 = No Orders Received
GO : 1 = Game Over
ST : 1=Strategic, 0=Tactical
VD : 1=Victory, 0=Defeat
DR : 1 = Draw
TD : 1 = Technical Difficulties
Block : Blockhouse Status
(Number of Hits)

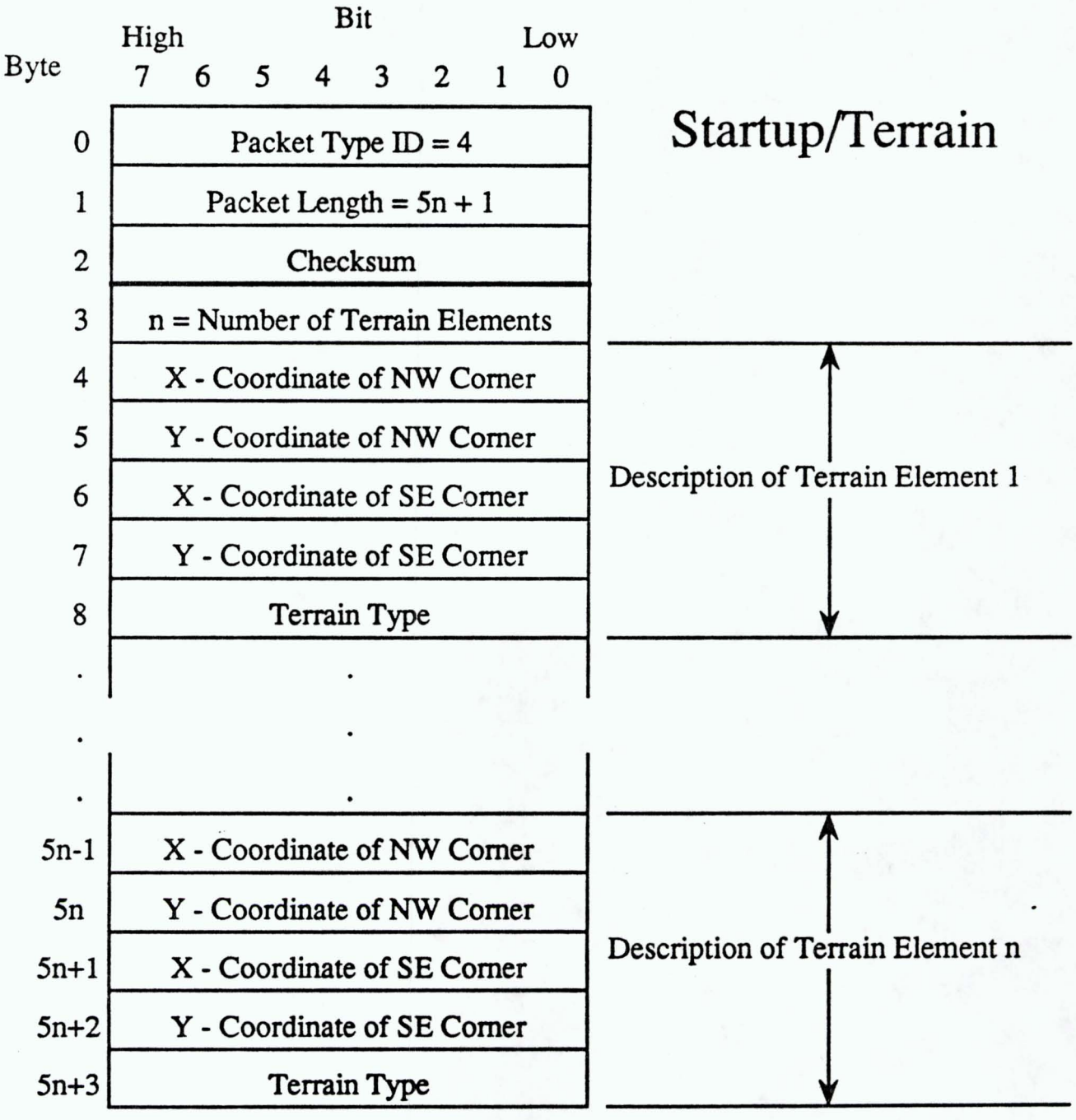
M : 1 = Tank is still mobile
W : 1 = Tank's weapon is still operational
A : 1 = Tank still has ammunition
C : 1 = Collision occurred
S : 1 = Tank was hit by a shell
R : 1 = Tank's weapon is reloading
MI: 1 = A movement order was ignored
WI: 1 = A weapon order was ignored

Object Report

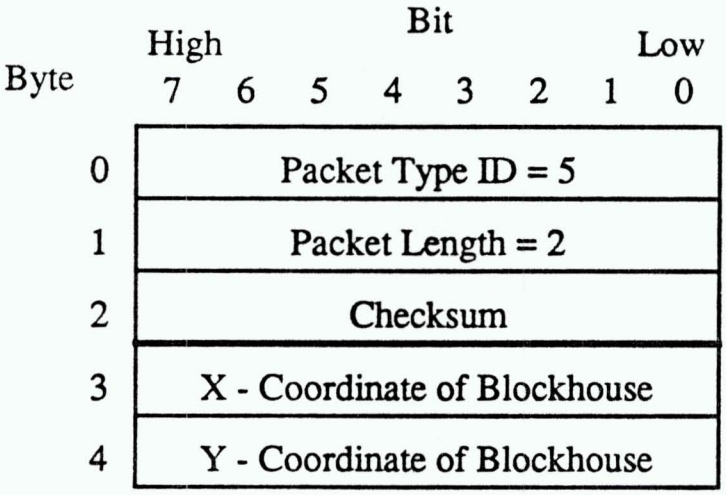


F : 1 = This tank just fired its weapon
T : 1 = This tank/blockhouse is being tracked
M: 1 = This tank is mobile
W: 1 = This tank's weapon is operational

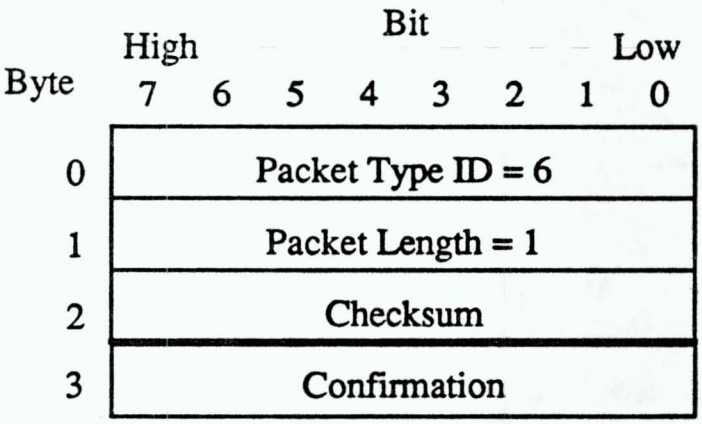
Obj ID:	**Bit Pattern:**
Enemy Tank	01
Enemy Blockhouse	10
Smoke	11



Terrain Types: 1 = Forest
2 = Water
3 = Mountain



Blockhouse Location



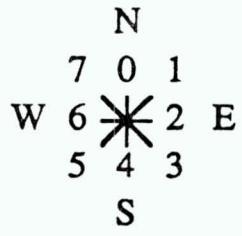
Terrain Acknowledge

Note: Confirmation is Zero if the competitor has received the Startup/Terrain message correctly (or if the competitor does not wish to verify the Startup/Terrain message.) Confirmation is NON-Zero if the competitor has received a malformed Startup/Terrain message and needs the simulator to resend it.

Surrender

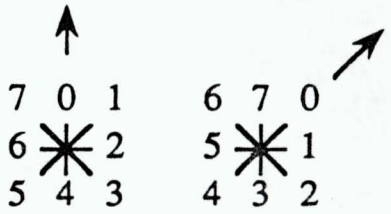
Byte	Bit: High 7	6	5	4	3	2	1	Low 0
0	Packet Type ID = 7							
1	Packet Length = 9							
2	Checksum							
3	'S' (83 Decimal)							
4	'U' (85 Decimal)							
5	'R' (82 Decimal)							
6	'R' (82 Decimal)							
7	'E' (69 Decimal)							
8	'N' (78 Decimal)							
9	'D' (68 Decimal)							
10	'E' (69 Decimal)							
11	'R' (82 Decimal)							

Note: This packet includes the ASCII string 'SURRENDER' in order to minimize the probability of mistaking a non-surrender packet for a surrender packet.



Heading:	Bit Pattern:
North	000
Northeast	001
East	010
Southeast	011
South	100
Southwest	101
West	110
Northwest	111

Speed:	Bit Pattern:
Ahead Full	10
Ahead Half	01
Halted	00
Back Half	11



Turret Facing : (Relative to Tank)	Bit Pattern:
Ahead	000
Ahead Right	001
Right	010
Back Right	011
Back	100
Back Left	101
Left	110
Ahead Left	111

In any **mask**, each bit represents a tank, with bit 0 representing the blockhouse and bits 1-7 representing tanks 1-7.

Terrain Type:	Bit Pattern:
Forest	00000001
Mountain	00000010
Water	00000011

Object ID:	Bit Pattern:
Enemy Tank	01
Enemy Blockhouse	10
Smoke	11