

Hostile Territory - design document
<http://h-territory.sourceforge.net>
first release: 8th of august, 2003, version 1.00
this release: 4th of february, 2004, version 1.41

TODO

- search for TODO-s and resolve them
- name of aliens: Telian

About this document

This is the design document of the game Hostile territory. Its purpose is to describe the game in its final state. There may be prior releases with less or different functionality. This document is not intended as an implementation guide as in some of the topics it will not go into sufficient detail required for coding.

Document history

- Version 1.4 – 4th of february, 2004:
 - major revision of the document
- Version 1.3 – 3rd of february, 2004:
 - cut some features to enable a concise design
 - first version that can be seen as a complete specification
- Version 1.2:
 - changed format to \TeX
 - revised document
- Version 1.1:
 - restructured document
- Version 1.0 – 8th of august, 2003:
 - created document

Design document

Contents

1	Game overview	5
1.1	What is the game about?	5
1.2	Background	5
1.3	Game type, categorization	5
1.4	Common questions about the game	5
2	Feature set	5
3	The game world	6
3.1	Terrain	6
3.2	Game parties	6
3.2.1	Characters	6
3.2.2	Squads	8
3.2.3	Space ship	8
3.2.4	Equipment	9
3.3	Radio communications and issuing commands	10
3.3.1	Available commands	11
3.4	Miscellaneous game objects	16
3.4.1	The artifact	16
3.4.2	Retreat point	16
3.4.3	Drop-off point	16
4	User interface	17
4.1	Overview	17
4.2	Issuing commands	18
4.3	Return values	18
4.4	Layout manager	18
4.5	User interface items	18
4.5.1	World map	18
4.5.2	Mini-map	19
4.5.3	Command buttons	19
4.5.4	Status report	19
4.5.5	Character display	20
4.5.6	Squad display	20
4.5.7	User interface commands	21
4.5.8	Out-of game commands	21
4.5.9	Message field	21
5	Music and sound	22

6	Graphics	22
6.1	World map	22
6.1.1	Character rendering	22
6.1.2	Equipment rendering	22
6.1.3	Terrain rendering	22
6.2	Mini-map	22
6.3	Command buttons	22
6.4	Status report	22
6.5	Characters display	22
6.6	Squad display	22
6.7	User interface commands	22
6.8	Out-of game commands	22
6.9	Message field	22
7	Artificial Intelligence	23
7.1	Events	23
7.1.1	Spot enemy	23
7.1.2	Unit attacked	23
7.1.3	Spot artifact	24
7.2	Priority	24
7.3	Retreating	24
7.4	Goals	24
7.4.1	Obey commands	24
7.4.2	Search and destroy	25
7.4.3	Search for the artifact	25
7.4.4	Search for enemies and hide	25
7.4.5	Search and hide	26
7.4.6	Hide and recover	26
7.4.7	Find allied units	26
7.5	The soldier's AI	26
7.6	The leader / squad leader AI	27
8	Game mechanics	28
8.1	Overview	28
8.2	Typical game run	28
8.2.1	Game states	28
8.2.2	Game turning points	28
8.3	Issuing commands	30
8.3.1	Action queues	30
8.4	Timing mechanics	30
8.4.1	Game control commands	30
8.4.2	Time of command execution	30
8.4.3	Turn order	30
8.4.4	Player turn and enemy turn	31
8.4.5	End of turn	31
8.4.6	Possible actions of a character during an enemy turn	31

9	Single player game	32
10	Multiplayer game	32
10.1	Game modes	32
11	Terrain editing & creation	32

1 Game overview

This section of the document should provide you with enough information to be able to decide whether you are interested in joining development of the game. It does not go into detail about anything, it just tries to be a short summary in a few pages.

1.1 What is the game about?

The player takes the role of a cybernetically enhanced soldier dispatched on a planet of a nearby solar system. She commands a squad of soldiers with a mission to retrieve an artifact and return it to their space ship.

1.2 Background

The game is set in the distant future, where mankind lives in peace with an alien race called the Telians. Their civilization stretches mainly on Earth and Mars. The world is ruled by a relatively weak government and several large corporations, who have comparable autonomy to that of a sovereign state today. Since about ten years, the possibility of a new alien threat has emerged. While it is kept secret from the population, the government and the corporations are already taking precautionary measures. One of these is the retrieval of the artifact, which is supposedly of alien origin.

1.3 Game type, categorization

The game is a tactical combat game. The player commands her avatar directly, other units are only issued commands via the avatar. Therefore it is possible for units to get isolated, return false information about themselves, etc. Interaction with the game world is possible for command purposes, such as taking cover, etc. The game can be played in a multiplayer mode as well.

1.4 Common questions about the game

insert them here

2 Feature set

- combat-strategy game
- isometric view of world
- multiplayer capabilities
- interaction with the world for combat purposes

3 The game world

Throughout the game, the player and her assigned squad operate on the surface of the planet. This section describes the parts of the game world the player can get in contact with.

3.1 Terrain

The artifact lies on a rocky, uninhabited planet. The terrain is either randomly generated on startup or created with a terrain editor (see 11). If randomly generated, the amount of open fields and maze-like canyons should be balanced. The artifact should not be placed on a large open field, where it is easy to spot. The terrain should be stored as a fine-grained height map with the possibility of placing individual objects (e.g. rocks) “on top” of the terrain. This gives enough flexibility to describe a rich variety of landscapes.

3.2 Game parties

Active elements in the game are characters, squads, equipment and space ship. Below is a brief outline of each category. If necessary, differences between the player's, the other corporate and the alien team are emphasized.

3.2.1 Characters

Characters are able to:

- move on the terrain
- take cover behind terrain features
- pick up equipment
- use equipment (for example, weapons)

3.2.1.1 Stats and skills

Characters have stats and skills:

- strength
- speed
- health
- perception
- close combat skill
- ranged weapons skill
- medical skill
- hiding skill

3.2.1.1.1 Skill test There are some more difficult tasks characters can not always successfully do. In this case they need to take a skill test. Skill tests can be taken per se or against a target, which can be a stat or skill of another character (with which the character is engaged in close combat, for example) or an attribute of an equipment (to determine if the character has spotted the item held by another character, for example). The higher the skill of the character the greater chances has the character to be successful.

3.2.1.2 Character state Characters have a *state*, which consists of:

- their stats
- their hp
- their equipment
- the position of the character in the world
- their action queue - a list of all the commands and their attributes that is to be carried out by the unit
- their goal (see 7.4)
- their strategies (see 3.3.1.7.11)
- their loyalty (towards each team and each person they know) - characters can turn against their own team and become traitors!
- what *they think* others think about their loyalty (each team and each person they know)
- what others think about the position of the character in the world

3.2.1.3 Player The avatar of the player can move freely in the game world as any other team party. It is important that the avatar is the only “trusted” source of information to the player. This means, the player sees only what her avatar sees and she knows only what her avatar knows. This is important for example, if radio communication is unavailable, because in such cases the player does not get any feedback from her units.

3.2.1.4 Leader The leaders of the enemy team take the same role as the player does for the allied team: they give orders to the other members of the team. Therefore, they have a more complex AI than a regular soldier.

3.2.1.5 Soldiers Soldiers have a limited AI. The actions to carry out are determined using a priority-based system.

They store their orders in a queue called the action queue, execute them and then wait for further instructions. If an action of higher priority than that of the currently executing is to be carried out, for example taking cover while moving, the soldier can temporarily suspend its current action to carry out the one with higher priority. For more on this, see 7.2.

3.2.2 Squads

Soldiers can be assigned to multiple squads. Squads can serve as a user interface feature for the player to be able to instruct more units at once and also as an active gameplay element where units collaborate with each other. Squads can share common instructions. For more on this, see 8.3.1.1.

Squads can temporarily have zero members. Please see 3.3.1.8.2 for more information on this.

3.2.2.1 Squads without a leader Squads without a leader are just a user interface feature: its soldiers do not communicate or help each other, they simply can be given common instructions.

3.2.2.2 Squads with a leader Squads with a leader are active gameplay elements. They can be created by promoting a soldier to the leader of an existing squad. The units in such a squad will communicate with each other and usually, only the squad leader will communicate with the player (except if one of the soldiers thinks the leader is a traitor). The units will request help from each other if in trouble. The squad leader can give commands to his squad just as the player.

Note: If the squad leader is a traitor and the other squad members do not know this, it is possible, that he “fools” the player and gives another instructions to his squad than he tells the player he does.

3.2.3 Space ship

The space ship is considered to be somewhat outside of game territory. It is possible to transfer friendly characters and equipment to or from the ship. Any communication and transfer to or from the ship can solely take place from the *drop-off point* and nowhere else (therefore, the artifact has to be carried there to be able to transport to the spacecraft and win the game)! For communication with the ship, a long-range communication device operated at the drop-off point is necessary. For more on this, see 3.4.3. When on the ship, characters cannot do anything except regenerating. The space ship is able to provide limited support fire for the team or launch a spy satellite.

< TODO: detailed description of space ship >

3.2.4 Equipment

Equipments have:

- a type
- a location
- *possibly* a state (for example ammunition of a weapon)
- attached objects (for example silencer on a gun)
- a visibility value
- a usage-indicator

Types of equipment consist of two orthogonal values, source and purpose. Sources determine which races can use the equipment:

- strictly human (Telian) - can only be used by humans (Telians)
- human (Telian) - can best be used by humans (Telians)
- generic - can be used by everyone
- alien - can best be used by aliens
- strictly alien - can only be used by aliens

Purpose determine what the object is intended for:

- ranged weapon
 - ballistic
 - energy
- close combat weapon
 - physical
 - energy
- passive weapon (e.g. mines, grenades)
 - explosive
 - energy
- medkit
- radar
 - short range (handheld)
 - long range (larger device)

- communication device
 - short range (handheld)
 - long range (larger device)
 - magnetic analysator
- protective equipment (armor, protective suits, etc.)
 - armor
 - energy shield
 - radiation protective suit (heavy, also protects against chemical and biological weapons)
 - biological / chemical protective suit
- all else I forgot ...

3.2.4.1 Visibility Not all equipment carried by a character is always visible. For example, a small handheld radar is only visible to another units, when the character uses it. To address this issue, all equipment has a visibility value and a usage-indicator. If another character sees the character carrying the equipment, he tests his perception against the visibility value modified with the usage-indicator. For more information, see 3.2.1.1.1.

< TODO: detailed description of objects with object states and distinction between Allies, Enemy - corporate team and Enemy - alien team >

3.3 Radio communications and issuing commands

Note: This section is related not only to the game world but also to the user interface as commanding units is done from within the game world. Please keep this in mind while you are reading this part of the document.

Through the game, the player selects a unit she wishes to command. Her avatar then tries to initialize radio communication with the unit (which costs some of her time) and if succeeded, she receives a status report from the unit. The player can then issue a command to the unit. The command will be sent to the unit who can choose between the following actions:

- not to respond anything
- acknowledge the command (“Yes, Sir!”)
- resist the command (“Go to hell!”)
- request further clarification (“I can’t hear you! Come in?”)
- propose an alternate command (“Sir, that’s impossible! We should remain in cover!”)

Some of this replies requires a further radio transmission from the player's avatar to the unit (which can be sent with the *confirm alternate command* or the *repeat transmission* command. See 3.3.1.7.2 and 3.3.1.7.3).

It is also possible to select more units, in which case the player will get a status report about the soldiers as a team from the senior among the units. The commands will be transmitted to all units, but only the senior unit will respond, according to his preferences.

If any of the units has some complaint about the command, he will contact the senior officer. The only exception to this is when the unit thinks the senior soldier is not to be trusted. In this case he will try to contact the player's avatar or any other soldier he trusts. If he does not reach anyone, he will probably disobey the orders.

Also, it is possible that a soldier contacts the player's avatar. The reasons for this include:

- they think another soldier has become a traitor
- they come under heavy enemy fire
- they get heavily wounded
- they spot an enemy

3.3.1 Available commands

3.3.1.1 Issuing commands and return values The player issues commands to the units through her avatar via radio. After a command, such as *Attack* is issued, a target also has to be specified. This is called the argument of the command and *return values* serve this purpose. Please see 4.2 and 4.3 for a discussion on these topics.

3.3.1.2 Priority Each command has a priority that is also dependent on the current strategy of the unit. This helps the AI resolve what to do in conflicting situations, for example if the unit has an attack order and is fired at, etc. For more information, see 7.2.

3.3.1.3 Argument tracking The argument of a command can be a moveable object, such as a character. In the case that it moves, its position, which serves as the target of the command is reevaluated each turn. If it goes beyond sight, the target of the command will be its last-seen position.

3.3.1.4 Center of a set The center of a set of positions is calculated as the arithmetic average of all of the positions in the set.

3.3.1.5 Converting to position A character, an equipment or sets thereof can be converted to a position or set of positions by taking the position of each item and removing duplicate positions.

3.3.1.6 General commands

3.3.1.6.1 Set retreat point The retreat point serves as a target for retreating units. At the start of the game, the retreat point is the drop-off point. This command changes the retreat point.

Arguments: The command takes one argument. The argument can be a position, a character or an equipment. The argument cannot be a set of characters or equipment. If the argument is a set of positions, the center of the set is taken as the argument. If the argument is a character or an item, argument tracking occurs.

3.3.1.7 Commands for single characters

3.3.1.7.1 Cancel all commands This command empties the action queue of the character. Takes no arguments.

3.3.1.7.2 Confirm alternate command This command confirms the alternate command proposed by a character. Only to be used when a unit proposes an alternate command as the reply to a command.

3.3.1.7.3 Repeat transmission This command repeats the last transmission. Only to be used when a unit requests further clarification or retransmission.

3.3.1.7.4 Move to The command instructs the unit to move to a destination.

Arguments: The command takes one argument as the destination. The argument can be a position, a character or an equipment. If the argument is a set of positions or items, further processing occurs. First if the argument is not a set of positions, it is converted to it. For each coherent region of positions, the medians of the two shorter sides is stored into the action queue as destinations of subsequent move commands. If there are non-coherent positions that are closer to each other than x

< TODO: determine x >

units, they are treated as a coherent region. Otherwise, they are added to the action queue of the unit as destinations of subsequent move commands.

If the target is a character or equipment, or sets thereof, argument tracking occurs.

< TODO: resolve this > Note: It should be evaluated, whether non-coherent *sets of positions* are able to be passed as a return value, for example when selecting numerous characters with a similar method, where the position of a character is not one single position. If they are, a somewhat more complicated (and computationally expensive) algorithm is necessary for handling these cases.

3.3.1.7.5 Attack The command instructs the unit to move in fire range and attack the target.

Arguments: The command takes one argument as the target. The argument can be a character, an equipment, a position or sets thereof. If the target is a set of positions, the center of the set is taken as a target. If the target is a character or equipment or sets thereof, argument tracking occurs.

Example: Two items of equipment are targeted. If they are at the same spot, two attack instructions with the same target position are issued. If in the next turn, they are taken up and moved to separate directions, the targets will be updated to reflect this change.

3.3.1.7.6 Retreat The command instructs the unit to try to retreat. The command takes no arguments. The AI determines which way the unit should move to avoid enemies and reach the retreat point. For more information, see 7.3.

3.3.1.7.7 Hold position The command instructs the unit to stay in place.

Arguments: The command takes one argument as a destination. The argument can be a position or a set of positions. If the argument is a set of positions, the center of the set is taken as the argument.

3.3.1.7.8 Guard another unit The command instructs the unit to follow and guard another unit or equipment. If a unit attacks the guarded unit, the guarding unit will attack it regardless of its *firing strategy* (see 3.3.1.7.11). If the unit guards equipment, the guarding unit will also attack an enemy if it approaches the equipment.

Arguments: The command takes one argument as the guarded unit(s). The argument can be a character or an equipment or sets thereof. If the guarding unit has to guard a set of objects, it tries to remain in the center of the set of their positions, if possible.

3.3.1.7.9 Get equipment The command instructs the unit to obtain an item. The unit will try to find and get it. If the desired equipment is seen at an enemy soldier, the unit may attack it if its *state 3.2.1.2* is appropriate.

3.3.1.7.10 Use equipment The command instructs the unit to use an equipment. If the unit does not have the equipment, a “Get equipment” command with the target type of equipment is inserted before this command in its action queue.

Arguments: The command takes one equipment as an argument. Optionally, if the use of the equipment needs a further argument, the command can take more arguments (e.g. the command to use a medkit on someone takes one additional argument).

3.3.1.7.11 Change strategy The command changes the strategy of the unit. The strategy of a unit consists of its:

1. firing strategy
 - fire at will
 - return fire
 - hold fire
2. moving strategy
 - roam
 - maneuver
 - hold position
3. communication strategy
 - verbose
 - obtain necessary information
 - communicate only important events
 - silent
4. artifact strategy - what the unit should do when it spots the artifact
(Note: this is considered an *important event*)
 - destroy (probably not the best idea as the player loses the game when this happens)
 - secure area
 - hide
 - retreat without artifact
 - retreat with artifact
 - ignore artifact

3.3.1.7.12 Set Goal This command sets the style the unit will behave when it has no commands to carry out. For more information, see 7.4. There are the following types of goals:

- obey commands: listen for commands of the player
- search and destroy
- search for the artifact
- search and hide
- hide and recover
- find allied units

3.3.1.7.13 Declare as traitor This command declares a soldier an enemy. From further on, he is considered hostile by the other soldiers (if they are loyal to the player) and he is excluded from radio communications.

3.3.1.7.14 Declare as suspicious This command declares a soldier to be suspicious. This state is between that of a team member and an enemy. Soldiers (if they are loyal to the player) will not attack the suspicious character, but they will not (easily) believe him nor follow his orders. They will not communicate with the suspicious soldier via radio.

3.3.1.7.15 Declare as teammate This command declares a soldier who was considered an enemy so far, a team member. From further on, he is considered friendly by other soldiers (if they are loyal to the player). He is also communicated with via radio.

3.3.1.8 Commands for a squad of selected units Besides the commands for single characters, there are commands which apply to a squad.

3.3.1.8.1 Promote to leader This command promotes a character to a leader of the squad. If the squad already had a leader, the old leader will be a regular member of the squad after this command.

Arguments: The command takes a character and a squad as an argument.

3.3.1.8.2 Add to squad This command adds character(s) to a squad. All commands of the squad that are in progress will be shared by the new squad members.

Arguments: The command takes a character or a set of characters and optionally a squad as an argument. If the squad argument is not specified, a new squad is created. The command can also take a position, an equipment or sets thereof as the first argument, in which case the members of the squad are reevaluated each turn and are those who stay at the designated position or hold the equipment. Note: This can cause a squad to temporarily have zero members. However, the squad has these items (positions, equipment) as special members and therefore it is not discarded and it will be fully functional when it has members again.

3.3.1.8.3 Remove from squad This command removes character(s) from a squad. All commands given to characters as members of the squad will be emptied from their action queue.

Arguments: The command takes a character or a set of characters and optionally a squad as an argument. If the squad argument is not supplied, the character(s) are removed from all of their squads.

3.3.1.8.4 Disband squad This command disbands the squad. All commands given to characters as members of the squad will be emptied from their

action queue.

Arguments: The command takes a squad as an argument.

3.4 Miscellaneous game objects

3.4.1 The artifact

The artifact is an alien weapon which probably got damaged. If so, it will hurt anyone without appropriate protection in its vicinity. The type of the weapon can be atomic, biological, chemical, psychic or a combination of these. The type and damage done to the weapon should be calculated at the start of the game as it will probably affect its surroundings (e.g. leaking radioactive or chemical material).

< TODO: describe in detail >

3.4.2 Retreat point

The retreat point serves as a target characters head to if they are ordered or need to retreat. At the start of the game it is at the location of the drop-off point.

3.4.3 Drop-off point

Due to the presence of a weak magnetic storm in the system it is not possible from any point of the surface to communicate with or initiate transfers to and from the ship. Units on the surface have to find spots where it is possible to overcome the interference caused by the storm with a long range communication device. As the storm itself always changes, the drop-off points do as well. There may be more than one drop-off points and there may be a system in how they change. There may be identical or different drop-off points for the different teams as their space ships are at different locations. It may be possible to predict where drop-off points will be (although not necessarily accurately). It is possible to detect where *current* drop-off points are (with the help of a *magnetic analyser*, see 3.2.4).

The drop-off point is an important element of gameplay. Only from the drop-off point is it possible to communicate with the ship (with the help of a long range communication device at that position) and only from and to this point can units, equipment and the artifact be transported.

See background.html for more information on magnetic storms.

4 User interface

4.1 Overview

The player commands her avatar. She issues commands to and receives feedback from other units through her, e.g. via radio or looking at a screen, etc. This makes propagation of unaccurate or false information and isolation of communicating parties possible.

The user interface consists of various items. Some of these items can be clicked at and they return a value. The items are placed on the screen by the *layout manager* and are listed below:

- world map
- mini-map
- command buttons
 - for selected units
 - for using equipment
 - game control commands (e.g. end turn, set action priority)
- status report
- characters display
 - allies
 - enemies (only if spotted)
- squad display
- user interface commands (for hiding and showing elements of the interface)
- out-of game commands
 - chat
 - multiplayer commands
 - save
 - load
 - quit
 - pause
 - options
 - credits
 - etc
- message field

4.2 Issuing commands

Through the game, the player selects a unit she wishes to command. When radio communication is established with the unit, the contents of his report and the possible commands to issue appear on the screen in a graphical format (e.g. hit points of the soldier, an attack button, etc). The player then can issue a command by selecting a button and specifying a target if needed. The target can be acquired from any user interface item that can return a suitable value (see 4.3 for more information).

It should be considered whether arguments can be selected autonomously from commands by always selecting a “first argument” with the left mouse button (if not clicked on a command) and a “second argument” with the right mouse button or if the arguments can only be specified after choosing a command.

The commands of the characters are stored in an action queue. A new command can be put in front or at the end of the action queue. Priorities can also modify where commands go in the queue.

4.3 Return values

If the player needs to specify an attribute of a command, he selects an object on a user interface item. Each item supporting selection of an object has return values. If the item offers one or more return values suitable for the command, the command chooses the most appropriate of them and uses it as an argument.

4.4 Layout manager

The layout manager arranges user interface items on the screen. It is responsible for knowing which items are visible and placing all of them appropriately (e.g. in a way that they do not overlap).

4.5 User interface items

4.5.1 World map

The world map displays the terrain, the location of the soldiers and items. The uncertainty of information is also indicated on the map.

Example: The player’s avatar has contradicting information about the place of one of her soldiers. The soldier is thus displayed on the map *two times*, both of which are opaque to indicate the uncertainty of the positions. If the player clicks on either of them, the unit will get selected.

Return value: The world map can return the following values:

- a position
- a character
- an equipment
- a set of positions (a rectangle)

- a set of characters
- a set of equipment

4.5.2 Mini-map

The mini-map is generated from the world map and displays the same information in a smaller scale.

Return value: The mini-map can return the following values:

- a position
 - a character
 - an equipment
 - a set of positions (a rectangle)
 - a set of characters
 - a set of equipment
- < TODO: determine size of the mini-map (in pixels) >

4.5.3 Command buttons

Each of the available commands (see 3.3.1) is displayed as a button with an image representing the command. When the player moves the mouse over the button, the name of the command appears on the screen. When the player clicks the button, the text remains on screen and get completed with what the player specifies as an attribute to the command.

Example: Character “z” is selected and the player moves the mouse over the move button, the text says: “z moves to...”. Then after the player clicks the button and a target on the map, the text changes to “z moves north” and fades out.

4.5.4 Status report

The status report should clearly display which unit or item is selected and display all of its characteristics. It is possible to make nested selections, for example select a soldier in a squad or an equipment of a soldier, or an equipment of a soldier in a squad, etc. There is a button inside the status report that cycles through the various possible states (for example, the various possible location for a character) in the case of uncertain information.

Return value: The status report item can return the following values:

- a position
- a character
- an equipment

- a set of positions (e.g. in the case of uncertain information or a squad)
- a set of characters (e.g. in the case of a squad)
- a set of equipment

4.5.5 Character display

The character display is a collection of buttons representing each known character. It should be indicated on the button which character it refers to. If the player moves the mouse over a button, the name and team of the respective character is displayed in the message field. Clicking the button selects the respective character.

Note: Enemy team members' buttons should only be available if they are in line of sight of an allied team member. If they are not, the button can point to their last-seen position. In this case they can not be selected by clicking the button (or perhaps the button can not be clicked). It is also important to indicate on the button if the information about the character is uncertain.

Examples for texts: "Allied::Mr. White", "Corporate::Soldier" or "Alien::Unknown".

Return value: The character display item can return the following values:

- a position
- a character
- a set of positions (e.g. in the case of uncertain information)
- a set of equipment

4.5.6 Squad display

The squad display is similar to the characters display. The buttons referencing the squads should indicate the designation of the squad and whether it has a leader. After clicking the button, the respective squad should be selected.

Return value: The squad display item can return the following values:

- a squad
- a set of characters
- a set of equipment
- a set of positions

4.5.7 User interface commands

The user interface commands are to communicate with the layout manager of the game. The two commands are “show” and “hide” and they both take items of the user interface as arguments. They instruct the layout manager to show/hide the respective UI item. It should be possible to manipulate only parts of the UI with these commands, such as showing only the attack button instead of all command buttons, but the convenient use of the command should be to show/hide all elements of an UI item.

Example: Clicking on the “show” and on the “command” buttons shows all commands. Holding down the mouse button on “command” for a short time brings up a pop-up menu where the player can choose specifically which commands to show.

4.5.8 Out-of game commands

Out-of game commands are New Game, Save, Load, Quit, Options and Credits which bring up a dialog box, Multiplayer chat, which allows the player to type into the message box and send a message, and Multiplayer commands which allow hosting/joining a game etc. None of these commands pauses the game (although some terminate/leave the game when they are executed). There is a planned pause command, please see 8.4.1.1 for more information.

4.5.9 Message field

This is the field where various messages, for example from units are displayed.

5 Music and sound

There should be ambient music throughout the game. The player should also be able to choose her own music. Perhaps a simple way for this is to include an mp3 player in the game.

6 Graphics

Note: This section is *very* incomplete for the moment, as I think it is essential to provide *at least* concept art (better the images which will be used with the final software) in a section about graphics. Unfortunately I cannot produce any that would be the quality I consider appropriate.

6.1 World map

The world map has an isometric view.

6.1.1 Character rendering

6.1.2 Equipment rendering

6.1.3 Terrain rendering

6.2 Mini-map

The mini-map is generated from the world map and displays the same information in a smaller scale.

6.3 Command buttons

6.4 Status report

6.5 Characters display

6.6 Squad display

6.7 User interface commands

6.8 Out-of game commands

6.9 Message field

7 Artificial Intelligence

AIs control the NPCs. It handles movement, combat, interaction with other (computer or human controlled) characters and response to certain events (injury, changes in environment, spotting an enemy, etc). AIs also decide on loyalty, that is whether to carry out their orders, go renegade or try to make a turn and accompany an enemy team.

As all of the characters have very similar possibilities what they can try to do, perhaps a unified AI can be used for all of them, with varieties for the major archetypes (and races) and small variations for individual characters.

Implementation note: perhaps it is a good idea to implement the AI as a separate object and let all characters query it with their statistics as inputs.

7.1 Events

Events are actions that the character takes or that are affecting it. Events have a set of responses from which one or more are chosen based on the unit's state (and other factors such as the presence of any nearby friendly units, for example). Except the "ignore event" response, all responses add a command to the action queue of the unit.

Events:

7.1.1 Spot enemy

The unit spots enemies.

Possible responses:

- attack the enemies
- hide
- retreat
- report to leader
- ignore event

7.1.2 Unit attacked

The unit gets under fire.

Possible responses:

- attack the enemies
- hide
- retreat
- report to leader

- if the attacking unit is not seen, try to trace it down
- ignore event

7.1.3 Spot artifact

The unit spots the artifact.

Possible responses: The response is determined by the appropriate strategy. See 3.3.1.7.11.

7.2 Priority

Each command has an assigned priority which is modified by the character state. If the unit receives an event, the response for the event is chosen according to the state of the unit. The response for the event is then assigned a priority and it is inserted into the action queue of the character according to that priority.

Example: A unit is attacked. A possible response for this event is to shoot back. However, the unit is returning the artifact to the drop-off point, which is assigned a higher priority than attacking the enemy, therefore he does not enter a fight until he reaches safety. If his firing strategy is set to “fire at will”, shooting back will probably have a higher priority so the unit might interrupt returning the artifact and engages the enemy.

7.3 Retreating

When a unit needs to retreat, it tries to find a way not blocked by enemy units to the retreat point. Depending on the unit state, it may attack some nearby enemy units if they are blocking its way.

7.4 Goals

Goals are broader heuristics that the AI applies. These are used by allied soldiers and squad leaders or by enemy leaders and soldiers. Leaders will issue commands to their soldiers and soldiers will carry out activities according to these principles. Each goal is described below together with the important events in the case that it is the active goal.

7.4.1 Obey commands

Listen for commands of the player. This goal can not be active for squad leaders, only for soldiers as it does not imply any self-sufficient behavior other than responding to events.

Important events:

- the unit is attacked
- the unit has killed an enemy

- the unit has spotted the artifact
- the unit has spotted an enemy
- the unit has recovered its health
- the unit has spotted allied units

7.4.2 Search and destroy

The unit searches out and attacks enemy units.

Important events:

- the unit is attacked
- the unit has killed an enemy
- the unit has spotted the artifact
- the unit has spotted an enemy

7.4.3 Search for the artifact

The unit searches for the artifact.

Important events:

- the unit is attacked
- the unit has killed an enemy
- the unit has spotted the artifact
- the unit has spotted an enemy

7.4.4 Search for enemies and hide

The unit searches out enemies but avoids conflict.

Important events:

- the unit is attacked
- the unit has killed an enemy
- the unit has spotted the artifact
- the unit has spotted an enemy

7.4.5 Search and hide

The unit searches for the artifact and hides if it spots enemies.

Important events:

- the unit is attacked
- the unit has killed an enemy
- the unit has spotted the artifact
- the unit has spotted an enemy

7.4.6 Hide and recover

The unit hides and recovers its health.

Important events:

- the unit is attacked
- the unit has killed an enemy
- the unit has spotted the artifact
- the unit has spotted an enemy
- the unit has recovered its health

7.4.7 Find allied units

The unit searches for allied units and joins them.

Important events:

- the unit is attacked
- the unit has killed an enemy
- the unit has spotted the artifact
- the unit has spotted an enemy
- the unit has spotted allied units

7.5 The soldier's AI

The soldiers are driven by a simple AI. They carry out whatever it is in their action queue. If they receive an event, they add the response for it in their action queue according to its priority. This can cause switching over from their previous task to the response. It is also possible to ignore specific events; in this case no action will be added to the queue of the unit.

7.6 The leader / squad leader AI

The leader AI is not responsible for the character of the leader itself (that is also governed by a soldier's AI), but rather gives orders to all of the squad / team members according to its current goal.

8 Game mechanics

8.1 Overview

This section describes non content-related information, such as game mechanics and typical game flow.

8.2 Typical game run

Informally, a game consists of the following stages:

1. setting up the team
2. start of game
3. one or more game turns
4. end of game

More precisely, the gameflow has a state which is altered by *game turning points* (see 8.2.2). The states of the game are:

1. game is not running
2. game is in progress
3. game has ended

< TODO: insert finite automaton of states here >

8.2.1 Game states

8.2.1.1 Game is not running The game is not running yet.

8.2.1.2 Game is in progress The game is in progress. See *Game turns* 8.2.2.3 for a description.

8.2.1.3 Game has ended After the game has ended, the data structures are not yet freed. A report may have to be saved and a new game with the same conditions may also have to be started.

8.2.2 Game turning points

8.2.2.1 Setting up the team The game plays on a foreign planet, therefore the player only has a predefined set of equipment available and no influence at all in choosing the soldiers. The set of equipment is deterministic, but the soldiers are generated randomly at least in part (some attributes might be deterministic). The point in setting up, or rather customizing the team is in deciding upon the equipment carried by the team members. The rest of the equipment is not lost throughout the game, as the player can request transfer of items from and to

the space ship.

After the player has set up the team, the next phase, *Start of game* 8.2.2.2 is carried out.

8.2.2.2 Start of game On the start of the game, a new random world is created. The game starts with the player and team members being on a predefined starting position (the initial *retreat point* and *drop-off point*, see 3.4.2 and 3.4.3), holding equipment specified by the player.

8.2.2.3 Game turns A game turn is the most important and most frequently executed part of the game. Therefore its code has to be fast and stable. The game turn is tied to the player's avatar. Each turn lasts a predefined time (e.g. 10 seconds). The player can issue commands to her avatar in this time-frame.

The opportunity of issuing commands to other units is not granted. If the player does not take time for her avatar to communicate with her men via radio, they will not receive any orders that turn. After the player finished issuing orders, she can inspect the situation and end the turn.

Realistically, a man has a fair idea of how long an activity like running, climbing, etc. takes, but usually this is not a precise value. We omit this inaccuracy, because it would be a pain playing the game if the character expects something to take 0.2 seconds and then it takes 3. The rationale behind omitting this is that the characters are trained soldiers so they practiced shooting, running, hiding, etc. long enough to know how long these activities take.

There are other factor that are not neglected, though. First, actions carried out with a different character state do not take the same time: they last longer if one is injured or carries heavy equipment. Second, there tends to be a slight variation in the amount of time an action takes even with no apparent differences in these conditions, which can be led back to reasons difficult or too cumbersome to model. Instead, the time of each action will be modeled with a probability distribution and specified with its expected value and deviation. With this calculating the time actions take can be done efficiently and precisely.

As the game is turn-based, it should be able to utilize the probably long waiting times while players think about their moves and calculate AI moves ahead to speed up gameplay.

8.2.2.4 End of game The game ends when:

- the player dies
- the artifact is destroyed
- the alien team gets the artifact and returns it to the team's spacecraft
- the enemy team gets the artifact and returns it to the team's spacecraft
- the player's team gets the artifact and returns it to the team's spacecraft

The player loses the game in all but the last case. After the game ends, a detailed report should be shown to the player (and saved to a file). It should be possible to restart the game with exactly the same conditions at this point.

8.3 Issuing commands - game mechanics issues

8.3.1 Action queues

8.3.1.1 Shared action queues of squads Each squad has an own action queue and commands to the squad are entered into this queue and then copied into all squad members' queues.

8.3.1.2 Set action priority commands Action priority commands can raise or lower an action's priority to move it in the queue. Forms of this command are +1, -1, set to infinity, set to negative infinity, whereas infinity and negative infinity begin from a higher value than what can be assigned to any action under normal conditions and increase (decrease) by one after each use of the command (also the +1 and -1 cases). This guarantees that priorities set to (negative) infinity will always have the highest (lowest) value, even higher (lower) than priorities set before.

8.4 Timing mechanics

The game is turn-based.

8.4.1 Game control commands

8.4.1.1 Pause command The command pause works as its name implies in single player mode. In a multiplayer game, it sends a request for pause to the other players (they receive it as a text message, which is displayed for at least a few seconds in their message fields) and they have to accept it by selecting the pause command themselves or decline it by not doing anything. The request times out in one minute.

Note: As you might have guessed, this feat is completely useless and will not be implemented until the game is turn-based. It is kept here for later use ;)

8.4.2 Time of command execution

The commands of the game are executed as soon as the player (or the AI) issues them to her avatar. Simultaneous command execution would not be practical because of the fact that the player issues commands to other units through her avatar.

8.4.3 Turn order

The turn order can either be static, random or dynamic. The first two options do not need further explanation. Dynamic turn order means it is calculated

at the beginning of each turn according to the actions of team members in the previous turn(s).

8.4.4 Player turn and enemy turn

The player can issue commands to her units in the player turn just as the enemy does in his turn. For more information, please see 8.2.2.3.

8.4.5 End of turn

This command ends the turn of the player. After the player ends his turn, any teams that have not yet taken their turns do so and the next turn begins with the first team in the turn order.

8.4.6 Possible actions of a character during an enemy turn

Characters are able to respond to events also during an enemy turn. They use up the same timeframe as in their own turn, so if they do not have any left, they cannot do anything. Note that it is possible for a unit to use up all its time for reactions of events *before* its turn is due. In this case it will not be able to do anything in his own turn (for example if a soldier is fired at by three enemies and decides to take cover, he won't have the opportunity to do anything before reaching safety). Reactions of the characters on events are determined by the character state.

9 Single player game

Any special comments on single player game come here ...

10 Multiplayer game

Multiplayer gaming on separate machines should also be possible. In this case, the players each are leaders of the teams. Other characters will remain computer-controlled, so the possibility of making allies with them etc. is still present.

10.1 Game modes

Besides playing the same mission as in a single player game, there will be other game modes available, like deathmatch, etc. It should be possible for players in these special modes to take the role of a soldier, thus enabling fully-human teams.

11 Terrain editing & creation

The game world is stored in a standardised format (XML). Perhaps there will be a utility which converts between a binary and a textual format, if there are performance reasons for this. The standard representation assures easy development of a terrain editor.