

# LIBERATING THE WORLD (Made Simple)

*Stan Shebs*

Department of Computer Science  
University of Utah

## ABSTRACT

This is an in-depth document on version 5 of the *xconq* family of programs. Version 5 is quite different from earlier versions; players familiar with those should read everything in here, especially the section on important changes. All aspects of play are covered. Details on customization may be found in a companion document.

*War is a matter of vital importance to the State; the province of life or death; the road to survival or ruin. It is mandatory that it be thoroughly studied. -- SUN TZU (ca 400 BC)*

Welcome to *xconq*, a chance for you to free the world from domination by evil empires. It is similar to previously distributed empire-building games, but with many more features. The same basic game is available with several different interfaces. *Xconq* is a fully X-based multi-player game, allowing almost any combination of human and machine players, and opening up remote X windows as necessary. There is also a restricted variant *cconq* that needs the curses terminal package only, but allows only one human player.

In the standard game, you start with one city and no knowledge of the world beyond your immediate vicinity. You must then explore, contact, and win wars against all the other players, who are trying to do exactly the same things to you! This is made harder by the limited information that the game supplies; except for the vicinity of your own possessions (and for certain type of units), the entire view is out-of-date, and you won't see enemies until they're close by.

The "standard game" is played on a small (usually 60x30 hexes) randomly generated map, against one machine player. Your first city will automatically start building your first military unit (usually infantry). When it is ready, the starting city will be overwritten by a picture of the unit, which is itself surrounded by a box-shaped cursor (the "unit cursor"). The mouse (or standard Unix direction keys) may then be used to designate any location for that unit to move to. This movement may take several turns, or the unit may stop before it gets there, usually because it is adjacent to something unfriendly. To attack, just direct the unit to move into a hex that shows another unit, and see what happens (flashes, and maybe a notice at the top of the screen). When you capture some kinds of units (usually cities), *xconq* will ask you what sort of units you want that unit to produce; '?' will display the possibilities.

In general, '?' will always work. When typed during normal movement, you will get a series of help screens, including commands, news, and unit characteristics. (This info may be written into printable files if the interface doesn't have the screen space necessary.)

The foregoing is sufficient to play - just jump in and go! After a few games, it should be clear what your units can and cannot do. The game will end automatically when your opponents are no longer capable of winning (either they have nothing left or they have given up). The following sections contain many boring details, and should be referred to for answers to questions.

## DEFINITIONS OF TERMS

An *xconq* game involves several *sides*, each of which has a human or machine player associated with it. Sides may be enemies, allies, or neutral with respect to each other, often start out in a hexagonal *country*.

A side owns a number of *units*, which includes the cities and the armed forces of that side. Units may also be *neutral*, and belong to no side (this is different from being on a neutral side). Units may be inside other units, in which case the one inside is in *occupant*, and the other is a *transport* (even if it can't move). Units always have *orders* that they follow, even when they appear to be under manual control. There are also *standing orders* that get passed to occupants automatically. The game is divided into a number of *turns*, during which each side gets to move some or all of its units. All the action happens over a *map* of a real or imaginary world that is divided into hexagonal shapes usually called *hexes*. Each hex has a *terrain* assumed to cover the entire hex. In some games, hexes also have a *populace* belonging to some side. Terrain on the map can produce *resources*, which are natural items ranging from water and food to gold and weapons; resources being carried by a unit are also called *supplies*. *Scenarios* are predefined games that set up maps, sides, units, and *win/lose conditions*, which define the circumstances under which one or more sides win or lose in the scenario.

The numbers and kinds of units, resources, and terrain are not built in; they are defined by a historical (or ahistorical!) *period*. This means that the following sections will be somewhat vague on specific units and behaviors, since the information applies to times ranging from Ancient Greece to Star Wars. Later sections will describe some of the periods that have been developed so far; in addition, complete online help is available on the period in effect.

## THE WORLD

*Geography defines the background to conflict. Gold mines are usually in the mountains, far from the sea. Islands tend to be left alone, unless they are on a direct path somewhere else. A seacoast town can be strategically useless if its approach is through shallow water. Attention to terrain and its effect on one's units can make the difference between winning and losing.*

The world map on which you play is a cylinder of variable height and diameter. Although it is always displayed as a rectangle, you can actually circumnavigate the world. The most northerly and southerly rows of hexes are out of bounds. Sizes can range from 20x20 "quicky" maps to the Earth at 1 degree resolution between 60 north and 60 south - no less than 360 by 120 hexes! When starting up, you have the choice of several maps of real areas, depending on the period, or by default you get a randomly-generated 60x30 map. You can get other sizes from about 10x10 up to whatever your machine's memory can hold, by using the **-M** option on the command line. The **-m** command line option loads a named map, and the **-x** option may also offer a menus of maps to use. Predefined maps usually have their own documentation, which is displayed on one of the help screens.

Each individual hex of the world contains one kind of terrain, which is assumed to more-or-less cover the entire hex. The exact set of terrains depends on the historical period; the set below is from the standard period, and is actually shared by many periods. Monochrome *xconq* uses icons for each type of terrain, which cannot possibly be described verbally, so use the help commands to decipher them.

Sea (dark blue) is assumed to be deep enough for any ship. Armies can't walk on water.

Shallows (light blue or cyan) include shoals, reefs, rivers, and any other sort of shallow water. They restrict movement of ships and may entirely prevent passage of the largest ships.

Plains (light green) are generally flat and hospitable areas. They usually offer no impediments to movement.

Forest (dark green) is dense forest or jungle, and may hinder movement for some kinds of units.

Swamps (gray) are half water and half land, and impassable to almost everybody.

Desert (yellow) ranges from Saharan sands to Sonoran cacti. It is always inhospitable but fast to move through - think of armor in North Africa.

Mountains (brown) are relatively barren and at higher elevation, thus are also inhospitable to troops.

Ice (white) is deep snow, ice, and glaciers. Only specially equipped ground units can pass over it, although most aircraft can fly over.

Vacuum (black) is outer space, included for the purpose of doing futuristic periods.

Each hex is adjacent to six others, and there is no special border to cross. By default, hexes represent areas about 100 km on a side, although many maps have larger or smaller scales.

Randomly generated maps vary in their "roughness", and in the percentages of each kind of terrain. These properties also depend on the period, and it is worthwhile to have a general idea of the values. Percentage coverage is simple (for instance, the earth is 70% covered by water), but roughness is more subtle; essentially the "jagginess" of the terrain. Very rough terrain has lots of sharp peaks and small islands, while smooth terrain has large flat continents.

## SIDES

*Politics provides the context for war; the war being a result of failed policy. The leader of a country is faced with the problem of achieving certain goals, either externally- or self-imposed. Diplomacy can sometimes accomplish the desired outcome, and is much cheaper to boot. When it fails, one country or another declares war, and any alliances immediately broaden its scope. Declaring peace again is much more difficult...*

Sides in the game can be allies or enemies in various combinations. Any two sides can form a formal alliance; human players do it by sending the message "alliance" to each other using the message command (see below). Machine players are "aware" of their relative incompetence, and will usually ally with each other (except in the case of a machine player attached to a display, so as to facilitate debugging). Players may become neutral or declare war by sending the messages "neutral" and "war" to another side. Scenarios may sometimes set up particular patterns of alliances, although there is nothing to prevent the players from changing them around during the game. If all the sides left in a game are allied, then it automatically ends.

Some displays distinguish alignments by using the same colors for allies as for yourself, while painting neutrals and enemies in distinctive colors. For others, you just have to remember who is on whose side.

Names of sides come from a scenario or are randomly generated from a list of names, depending on period. If the period doesn't define any names for sides, then the list will be 100+ contemporary names (including Botswanans and Peruvians). Players may also rename themselves, using a command (see below). Since it is usually hard to remember which player has which name, many mentions of sides include the display that the side is using (or nothing if a machine player), or sometimes the number of the side (especially for input).

When a side loses, for whatever reason, units are either destroyed or made neutral (depending on unit and period). In the standard period, infantry is destroyed, while cities become neutral (thus easy pickings for the remaining players who get to them the quickest).

Informal alliances frequently happen in games involving more than two people, so I have a few words of advice. First, an alliance between two of the players is almost certain in a three-person game, and inevitably results in the "odd man out" being quickly defeated. In four-person games, the alliances should be decided after looking at the map via "-v", so that one pair is not hopelessly separated. Five or more players is going to be a free-for-all of formal and informal alliances. Some scenarios are designed with a particular number of players in mind; hopefully they will also have some natural balance.

## UNITS

*War is based on the application of force to the other side, using whatever is available; from spears and arrows to the high-tech equipment available as a significant fraction of a nation's GNP.*

Units can be almost anything: armies, balloons, triremes, cavalry, battleships, bridges, headquarters, cities. Units move around, attack other units, produce resources, and build more units, among other things. Individual units occupy entire hexes, and no other unit can enter that hex unless it can occupy or be occupied by the unit already there.

Only some kinds of units can build other units, at a rate depending on the period, the unit being built, and the unit doing the building. The first unit that is produced takes somewhat longer, and the very first unit built by a side can take even longer (research and development time), but then succeeding units come out at a constant rate. There is no memory about production, so switching to a different type then switching back still incurs the extra startup time. Most units that do building will do it all the time, and only stop when explicitly directed to (such as cities), while others need to be directed to build, and cannot move while doing

so (such as engineers building a base).

Once created, a unit moves according to its orders, and subject to various constraints - armies can't swim, ships can't walk, etc. Units can sometimes be disbanded with a command (depending on the period), by losing them in battle, by running out of supplies, by being taken prisoner when a unit is captured, or by garrisoning a captured unit.

Every unit starts out with a number of *hit points* representing its strength, and possibly supplies of food, fuel etc that it carries around. Supplies are used up by movement, combat, and by just existing, and are created by production on certain terrain types, or by transference from some other unit.

There is only one situation under which several units can be in the same hex at once; if one is a transport of some sort and the others are its passengers or occupants. The notion of "transport" and "occupant" is general, and covers fighters on carriers, ships in port, bombs in bombers, and troops being led by a general. Occupants board by moving into the hex occupied by the transport, but will refuse to go if the transport is full or can't carry that type of unit. Getting on board takes a number of moves of that unit; if there are any left, it may move off or take some other action. Transports can also move over a occupant to take it on, but only if the transport can move on the terrain that the occupant is on. Occupants always move with the transport (that's what transporting is all about!), but may leave at any time if possible, either onto a valid terrain or onto another transport. To debark, just move the unit off (the cursor indicates that the occupant and not the transport is to be moved). Usually, you will want to put the occupant on sentry duty while moving the transport, and so must wake the occupants up before they can be moved again.

## THE GAME

Games may be predefined scenarios, which define the map, sides, and units, or they may be randomly generated. If randomly generated, depending on the period, you start with either a country-full of units or just one, which may or may not have its production defined already. If you start with one, the period may also define some neutral units in your country, which should be captured as quickly as possible.

Sometimes the map will be too small or have the wrong terrain, and then *xconq* will fail at placement and exit instantly. There is not much you can do at that point except to try again or relax the constraints, perhaps by reducing the number of sides or increasing the map size. (This can also be fixed by altering the period - see the customization document for details.) Since there is a lot of randomness in placement, second tries are frequently successful, although tenth tries usually indicate a real problem.

A turn consists of several phases, although only one actually involves player interaction:

### Spy Phase

Leakage of information from one side to another.

### Disaster Phase

Revolts, surrenders, attrition, and accidents.

### Build Phase

Construction of new units and repair of damaged ones.

### Supply Phase

Production and distribution of resources.

### Movement Phase

Automatic and manual movement of all units.

### Consumption Phase

Details relating to supply usage during movement.

During the movement phase, the program iterates through all units, prompting each side to give orders to any unit that is awake or becomes awake during the course of its move. One consequence is that you will not have a chance to change orders, look around, or do anything else if no unit produces a unit and no units wake up. This speeds playing but can be annoying if you get overrun and lose without ever getting a chance to respond (but do you deserve anything else for pursuing a "hands-off" management strategy?). Sides that lose are automatically cut out of the game. Since one additional iteration is needed to verify that somebody lost, the final winner will have to go through an entire turn before the game will exit (doing the sentry command on everything is easy and quick).

The game ends when the win/lose conditions have been met; these vary from scenario to scenario. For a randomly-generated game, the end comes when no mutual enemies are left, whether by elimination or by peace. Usually this means that only one side is left alive, but multiple machine players (not associated with displays - the usual case) are always allied, and thus may win as a group. This also means that a single member of the alliance will not resign until the position of the whole alliance is hopeless; after all, the WWII Allies included several brigades of Polish troops after Poland was overrun.

The last player must type a key to close down the windows (this is so that they will stay up for everybody to look at). When the game closes down, the winners (if any) will be listed. If the STATISTICS option has been set by the installer, *xconq* will write a file "stats.xconq" into the current directory. This file summarizes some crucial statistics concerning combat performance, losses, and other miscellany. It is quite useful for rationalizing your humiliating defeat!

## DISASTER PHASE

*War is inherently random. Both military and civilian units desert, get diseases, have accidents, defect, and surrender without a struggle. These effects cannot be eliminated completely, but can be reduced by keeping one's forces out of hazardous situations and by keeping morale up.*

Three types of disasters can befall a unit during the disaster phase: revolt/surrender, attrition, and accidents.

Revolts and surrenders are really the same sorts of occurrence; a unit changes sides spontaneously, perhaps to neutrality, perhaps to the side of a nearby enemy unit. During every disaster phase, each unit makes a revolt check. The revolt chance is a hundredth percentage. When a unit revolts, it changes to its original side (whatever the unit started out as - i.e. your initial units will never revolt). Occupants will either change over or be killed. Any construction will be cancelled, unless the scenario is one in which construction changes are not allowed.

Surrender only occurs if a unit is capable of capture is present. The capturing unit does not move. Occupants of the surrendering unit also change over or die. Chance of surrender is increased by low unit morale.

The chance of surrender can be greatly increased (depending on period) by surrounding the unit completely. This includes naval units for any sea hexes. One of the surrounding units must be capable (even if only a small chance) of capturing the unit by direct attack. The siege is only in effect in those turns where the unit is completely surrounded. When the unit surrenders, one of the "capture-capable" units will be randomly picked to accept the surrender, and things happen as for a direct assault (described below). Note that if several sides are surrounding the same unit, the selection is still random from among those sides, as long as the side is an enemy.

Attrition is a "slow death" process applicable primarily to multi-hp units. It takes away some number of hit points each time it occurs, and kills units only if they have no points left. Attrition is also specified in hundredths of a percent, and depends on unit type and terrain type. Morale drops by 1 when attrition occurs. A message will be displayed as well.

Finally, there is a chance for an accident to destroy a unit instantly and totally. Like attrition, this depends on both unit and terrain type, and is measured in hundredths of a percent. If the accident occurs, the unit is killed along with all occupants. A message will be displayed.

## BUILD PHASE

*Sustained efforts in a war depend vitally on the replacement and repair of forces destroyed or damaged in combat. In total war, the production base constitutes a chief strategic target, to be isolated or destroyed if possible. Repair of units is also significant since a battle may result only in damage, but be successful nevertheless if units must retire (as a cheaper alternative to new production). Historically, battle damage has resulted in the termination of an entire campaign.*

During the build phase, units construct new units and repair damaged occupants or transports (or themselves).

Construction is straightforward; the schedule is decremented once/turn. When it has counted down to zero, the unit is created, and placed either as an occupant of the builder, or the builder is made to occupy the new unit. If neither alternative works (perhaps because the builder is full already), then completion is

postponed, and attempted on the next turn. This will be repeated indefinitely. If the new unit cannot be placed at all, it is thrown away. If the period specifies that the builder is to guard the new unit, then the builder will be assigned to garrison the new unit, and is destroyed.

Repair happens automatically if the damaged unit contains or is contained by another unit capable of repairing, or if the unit can repair itself. The repair rate is depends on both the repairer and repairee, and can happen no faster than one hp/turn.

## SUPPLY PHASE

*The Allies floated to victory on a sea of oil. -- LORD CURZON.*

Resources themselves are basically inanimate material that come in varying amounts and are governed by conservation laws. They can be produced, moved around, and consumed during various activities. Resources originate either as supplies carried by units at the outset, or more typically, through production by units. Production rate depends on unit, resource, and terrain types, and is unaffected by side changes, combat, or anything else. Produced resources go into the producing unit's storage.

Excess production is discarded, unless it can be unloaded into the producer's occupying units, or distributed to nearby units via *supply lines*. Supply lines automatically exist between units that are close enough (as decreed by the period), and there is no need for explicit manipulation.

Units consume their supplies, both in the course of existence, and by motion/combat. The rate depends on period and unit type; it consists of an overhead consumed each turn without fail, and consumption for each hex of movement. The total is a max, not a sum, since units with a constant consumption rate are not likely to need additional supplies to move (consider foot soldiers who eat as much sitting around as they do walking). Supplies may also be consumed for production and repair, again depending on period and unit types, but this consumption happens during the build phase. Consumption is not affected by the situation of the consuming unit; armies in troop transports eat just as much as when in the field.

Supply line length depends on the period and the units on both ends, but is not affected by the intervening terrain. Supply redistribution is managed by logistics experts, who are ignorant of the war effort and seek only to even everything out. The redistribution method is rather adhoc; units try to get rid of all their excess supply, and try to take up supply from other units within supply range. Each direction is controlled independently, so for instance airplanes can get automatically refueled from a nearby city, but not from each other. No unit will transfer all of its supply via supply lines. Normally units in the same hex can exchange supplies, but some periods can disable this behavior, so that explicit transfer using the give and take commands is always necessary.

## MOVEMENT AND COMBAT

The movement phase is the one in which all the action happens. At its outset, the phase computes the number of moves available to each unit. This value is essentially the maximum of the unit's moves on each type of terrain. The movement phase continues until all these moves have been expended in some way or another.

Some periods may define a small chance of random movement, in which the unit moving actually goes in some other direction than that intended. This is a potentially dangerous occurrence, since the unit will be destroyed if the hex is impassable or contains another unit (whether or not the other unit can take the moving one as an occupant).

All combat occurs during the movement phase. Battles happen when one unit attempts to move onto a hex occupied by an unfriendly unit. In most periods, each unit attacks the other equally well, but if "counterattack" is not enabled, then the defender just has to sit and take the punishment. The outcome is determined independently for each unit, based on a probability table; this means that both draws and mutual damage/destruction are possible. The odds are the same whether a unit is attacking or being attacked. Ammunition may be expended by each unit in each combat - if the ammo is gone, then the attacker will not attack and the defender cannot defend itself. The results are announced both by a message and by some flashes on the screen (the size of the flash corresponds to damage seriousness). Damage is assessed using hit points, and if the hit points are zero, the unit is destroyed, along with any occupants. Typically armies have only one hit point each, so they are destroyed if hit. Units with multiple hit points may be *crippled* if their hit

points drop below a period-specified level. Crippled units move more slowly (in proportion to their damage), have reduced transport volume, cannot repair anything, and do not make progress on any construction. The final outcome of combat depends on whether the defender was destroyed. If so, the attacker will move into the defender's position (if possible), otherwise no movement will happen.

If a unit is hit sufficiently hard, that is considered a "nuke" and you get more spectacular visual effects, plus the hex is converted into desert or something else desolate.

Some units are capable of capturing other units, with a probability depending on the types of both units involved. If the capture attempt is successful, the capturer will move into the hex if possible, either as occupant or transport. In some periods, the capturer may be all or partially disbanded, to serve as guards. The regular attack as described above always happens first.

## ORDERS

*A perennial feature of the highest level of command is its inherent complexity. Although the use of subordinates reduces the bewilderment somewhat, the commander-in-chief must still keep in mind hundreds of apparently unrelated facts; the state of the weather, the past performance of units, the current goals of the war, and many other things. It is very important that lower-level units be able to operate on their own as much as possible.*

Although units have been said to "move", in actuality they follow orders, some kinds of which specify movement. When you are moving a unit hex-by-hex, it is following the order "Awake", which just means that every turn it asks what to do next. There are many kinds of orders. Some, such as movement in a given direction, or to a given place, take parameters, but all take a repetition, which tells for how many turns the unit will carry out the order. (For some orders, the repetition is not particularly meaningful, and is ignored.) Repetition is always specified as a prefix numeric argument to commands.

Orders that a unit can do include:

Awake        ask for a movement or other command.

Sentry       sit at the present location as long as the repetition says.

MoveDir     move in the given direction.

MoveTo      try to move to the given location.

FollowCoast  
              attempt to follow a coastline.

FollowLeader  
              move towards another given unit (which must be one of your own).

Patrol       go back and forth between two points.

Embark      embark on the first available transport. If no transport is available, it will just wait. Once it embarks, the remainder of its time is changed to doing sentry duty.

Most movement commands just give these orders to the unit currently being prompted about. In addition, a unit may be given "standing orders", which will be passed to any unit of a particular type entering or produced in that unit. This is useful for a variety of purposes, such as staging fighter planes up to the front lines or sending ships out on standard patrols.

## THE DISPLAY

*All warfare is based on deception. -- SUN TZU*

When a game is started up, it opens a number of windows, of which the most important is the area map (which therefore gets the largest window). Above the area map are several windows for status and notifications, and next to those is a turn counter and a list of all sides in the game. The middle right side has a list of all unit types, used for statistics display, while the lower right-hand corner has a map of the world (if the display is sufficiently large).

All of the *xcong* windows are actually subwindows of a main window with a patterned background you can see here and there. You can iconify and move the main window, and the subwindows will keep their

relative positions. The largest of these is the map, which is a (typically) 30x30 section of the world in full detail. The view is scrolled around as necessary (remember that the world is cylindrical).

To the right and down, you see a map of the whole world. This view is like the close-in map, but units and units are rendered as solid blobs, since the world is too large to permit any detail. To assist in matching up the two different views, the world map includes an outline box indicating the position of the close-in view.

Three text displays are stacked at the top of the screen. The uppermost is about ten lines of notices about various occurrences, each prefixed by the number of the turn in which it was issued. The display scrolls. The next four lines are an information window that summarizes the status of the unit or unit at the current cursor position. It can display info about enemy things also, but of course the amount of information is less. Finally there is a one-line prompt window just above the map, in which all questions and prompts appear.

The list of sides playing appears in the upper right corner. Sides that have already lost appear with a line through them, while the currently moving side has a "\*" next to it. Your own side name is highlighted or inverted. The name of the side, its host (if any), and the number of that side are shown. On color displays, the color of the number indicates the alignment of that side.

The X interface draws the area map as a number of hexagonal shapes with icons for units superimposed. Unknown territory is black. Your own possessions appear in black, neutral units in gray, and all enemies in red. If there is more than one enemy side, they are distinguished by the number of their side in the upper right corner of the icon. Not all enemy units will be visible; the chance of seeing one may be very low, or depend on viewing with the right type of unit. Monochrome screens display enemies and neutrals as inverted from your own appearance, and all enemy units/units have numbers, to distinguish them from the neutral units.

The curses interface displays each hex as two characters side-by-side. Terrain is a character representing the terrain, as are units. The second character in a hex is either the side number of an enemy, or an apostrophe for neutrals.

As mentioned previously, the view is a record of what has already been seen, but is not updated except in the immediate vicinity of your own units and units. In multi-human games, all screens will be kept up-to-date simultaneously, so that persons waiting for their turns can see enemy units moving around, units change hands, and so forth. The previous view of the hex is also remembered and will be described by moving the cursor to a hex in survey mode.

## INPUT

Input may be supplied both from the mouse and the keyboard. Moving the mouse cursor to a screen position and clicking either button has the effect of issuing MoveTo orders to the current unit, which will be carried out until successful. There are two exceptions. The first is that if the mouse is on the unit itself, the unit sits where it is until the next turn (same as the ' ' command below). If the desired new position is adjacent, the unit will unconditionally move there - this is useful for attacking enemies. At present, there is no special meaning attached to particular mouse buttons.

As an alternative to using the mouse, the standard directions (h = West, l = East, y = NW, u = NE, b = SW, n = SE) can be used to specify movement. Uppercase versions of these makes the unit move forever in that direction. Letter directions are really only of use when the mouse fails, or for diehard Unix game players for whom the direction keys have been permanently wired in the brain! Note that in a hexagonal system, 'j' and 'k' are not meaningful.

Any command may be prefixed by a single numeric argument (which may be positive or negative). Not all commands will use this number, while others need a number to know how often to repeat an order, or perhaps for some other reason. The "current unit" is the one being prompted about, while the "main unit" is the one occupying the hex itself (as opposed to its occupants).

Commands to give units orders typically default to a repetition of 100 turns (In some cases, this is meaningless, as in moving to a place):

- s Sentry; the current unit "goes to sleep", only wakes up by explicit command or when an enemy pops into view (the enemy unit won't necessarily be adjacent, if the unit on sentry duty can see far-off



hexes).

- w Wake up; the unit's orders will be erased (whatever they were) and it will ask about its next order (not always immediately). This command interprets an argument as a radius for the effect of waking up; for instance, the default of 0 means to wake only the unit itself, 1 means to wake up adjacent units as well, 15 will wake up a screenful of units, and 999 will usually wake all units.
- W Wake all; both the unit and all its occupants will be woken up, as well as all of their occupants, recursively. The command is otherwise identical to 'w'.
- Space Sit; unit goes on sentry duty for exactly one turn, and will ask for a move in the next turn. Useful for waiting one or two turns.
- r Return; unit returns to nearest transport by shortest route. It will not return to transports with no room or no supplies. Most useful for aircraft, but works with any unit.
- m Move to a position; this is equivalent to mouse clicking, but can be used with mouse-less interfaces or to move further than one screen width. You will be prompted to do movement commands (either mouse or keyboard), then can use the space bar to designate the final destination.
- f Follow leader; follow another unit. The program will ask you to designate a unit to be followed, which must be one of your own. The interaction is identical with that for 'm'. Each turn the unit either attempts to move towards its given leader, and sits if it is within a couple hexes of the leader. Units will not follow themselves.
- F Follow coast; follow a coast line. This can be applied to any unit, although the unit will immediately wake up again if it is not next to some sort of terrain that it cannot move into. The command will prompt for a standard direction to decide how to start, then a contour-following algorithm will continually try to keep the unit adjacent to impassable terrain (thus armor might use this command to go around a mountain range or forest). Because the terrain is in discrete hexes, it is possible for the unit to get confused, but that's life.
- Z Patrol; set the current unit to go back and forth between two points. One point is the unit's current position, and the other will be prompted for. The roundtrip will be repeated for the number of times designated by the command's argument.

Commands for modes. There are only two modes defined at present: move mode (the default) and survey mode. Most commands work the same in both modes.

- z Survey mode; toggle into/out of survey mode. In survey mode, movement pushes a cursor and allows you to look at things. The other commands are still available; for instance, you can give a unit new orders or to set unit production.

Commands for general manipulation of units:

- d Delay move; unit's movement is delayed until all your other units have been moved, then it will be prompted for again. Useful in crowded situations. Delay can be used on any number of units any number of times during a turn.
- P Set unit production; will ask for a type of unit (if a choice possible) and then schedule construction for a unit of that type. Any partial production will be discarded.
- ^P Set unit production automatically; will use the machine algorithm to determine the product.
- I Idle; cancel production for the given unit and leave it idle for a while Default is 100 turns, argument to command overrides.
- C Call unit by name; prompts for a string by which current unit will be referred to. If string is empty, unit name will be removed. If this command is done when the cursor is on an empty hex, the string will become the new name of the whole side instead.
- D Disband; unit disbands and disappears from game. Not all units can be disbanded; for instance, most periods will not allow the voluntary destruction of a city. If an occupant is disbanded, then its transport will get any available resources, both those held as supplies and any used in making the disbanded unit (possibly not all, depending on the period's "efficiency" parameter).

- a Cycle through occupants; this can only be used in survey mode, and allows examination of each occupant and its suboccupants. The order of traversal is depth-first, and cycles through all units in the hex repeatedly.
- x Mark a unit; used with embarkation, below. Only one unit is ever marked at any one time.
- e Embark; put the current unit onto a random transport in the same hex. This is useful when you don't want to move transport or unit out just for the purpose of boarding. If the marked unit is in the same hex, it will be used as the transport. May given argument to have unit wait that many turns for a suitable transport. Once it is on board, it goes to sentry duty for the remainder of its time.
- g Give supplies; transfer all types of supplies from the current unit to its transport, if there is one. The default is to try to fill up the transport if possible. If an argument is supplied, it means to transfer exactly that quantity of each resource type. If the current unit is low on some type, then it will transfer half of what was requested. (Repeating the command transfers half again, and so forth.)
- t Take supplies; transfer all types of supplies to the current unit from its transport, if there is one. The default is to try to fill up the current unit, or to interpret the argument as the quantity to take. If the transport is low, then it will only transfer half of what was requested.
- O Set standing orders; will ask for type of unit to which standing orders will apply, then goes into a "teach mode"; the next input will saved as an order rather than being applied to some unit. When any unit of the appropriate type enters the unit with the standing order, it will be given those orders and carry them out. There is no way to cancel standing orders at present, but occupants can be set to wake up during entry. A unit receiving a standing order will automatically leave automatic mode, unless the standing order puts it in automatic mode. All standing orders are displayed before the unit type is prompted for. If no change is desired, then exit the command by typing the escape key.
- G Give unit; give the current unit to the side specified by the argument. If the side is invalid, then the unit is made neutral. Not all unit types can be given away.
- ^A Automate unit; allows the machine to control this unit until it is explicitly woken by the user. New orders are only given after the current orders have expired, allowing the user to give order in survey mode. A standing order can be left to automate a unit. Such an order may be combined with another order, but must be left as the second order for the same unit type. If both a normal order and an automate order are present, the regular order is carried out and then new orders will be given by the machine.

Commands for side manipulation:

- c Center; the list of units is sorted so that the one at the current cursor will move first, and others move in concentric circles outward. This is useful for concentrating on one particular area and reducing the amount of map redrawing.
- M Message; send a message to another side. The side is specified by giving its number as a prefix to the command; if the number is not the number of a side, then your message will be broadcast to all sides (including yourself). You may type in a message up to the length of the prompt window. Backspacing is available. When a newline is entered, the message is sent immediately to the destination. Specially recognized messages must be typed exactly, with no other words or characters in the message:
  - war Declare war. Only one side need do this. This involves all allies on both sides immediately.
  - neutral Declare neutrality. Both sides must send this message to each other.
  - alliance If two sides send this message to each other, they become formal allies. The display changes to reflect this, as do things like wakeups, etc. All sides in the game will hear about the alliance.
  - briefing Sends the view of all of your units to the other side. Useful for allies, as well as to convince a victim that further resistance is hopeless. Of course, the victim's position might not be so hopeless after all, in which case you've given away all your secrets!
- . Recenter; put the hex with the cursor in it in the center of the screen.

**^F** Follow Action; toggle look mode. Look mode puts any changes in the view on the screen when it is not your turn. The mode window will be followed by a "L" in look mode.

#### Commands for game control.

- X** Resign; resign from the game, (asks for confirmation first). The effect is the same as losing.
- Q** Quit; terminate the game for everybody (asks for confirmation first). Note that although this can be used even in multi-human games, applying it without prior consent of the other players is definitely anti-social!
- S** Save game; record the game state into a file and exit (asks for confirmation first). The saved game is ASCII and unprotected, so it's possible to "checkpoint" games and engage in other kinds of cheating. The game exits once it has been saved. To restart, start up the program without any command line arguments, and in the directory where the save file is located. If players are specified on the command line, then they override the saved player data. This is one way to switch sides; for instance, saving from "xconq" and restarting with "xconq -r -e 1 \$DISPLAY" has the effect of you switching sides with the machine.
- A** Add player; add a new player to the game (not implemented yet).
- o** Options; set various options. Each option is a single character. Options at present include:
- g** Graph; toggle between text and bar graph displays about the current unit's supplies, hit points, etc.
- d** Display mode; cycle between four different color display techniques. The curses interface also has two display modes (one or two terrain chars per hex), but you still have to cycle between four modes.
- i** Invert; invert foreground and background colors everywhere (monochrome only).
- w** Width; set the width of the map display to be the value of the argument. This is measured in hexes.
- h** Height; set the height of the map display to be the value of the argument. This is measured in hexes.
- n** Notices; set the number of notice lines at the top of the screen.
- r** Robot; convert yourself into a machine player. This asks for confirmation. If there are no other human players, ^C is re-enabled, so you can terminate the program. This may be reversed by typing Escape. (Not functional in curses version.)
- m** Monochrome; This has the bizarre behavior of converting a color display into its monochrome equivalent. Actually intended for debugging, but pretty flashy if you're bored.
- b** Start Beep Time; A beep will sound at the start of your turn if it is more than this many seconds since your last turn.

#### Information commands.

- ?** General help; show a sequence of help screens, starting with a list of commands, then a display of icons, then any news, then general info about the period, then the characteristics of each unit (as for '=' below). You may page back and forth through the screens. This general help is available in both move and survey modes. Some specialized prompts (such as for unit type) will also recognize '?', but will only supply more details about possible answers to the prompt.
- /** Identify; display a short phrase indicating what is being seen in the hex at the current cursor position. This works in all modes, and is useful for deciphering unusual colors or icons.
- =** List the characteristics of a type of unit. It will prompt for the type, then format all the period-specific details into a semi-readable summary. To get a hardcopy of this, use 'p'.
- p** Print; dump all the characteristics of all unit types into a file "parms.xconq". This file may be printed, and is very useful for learning about a period. It will include designer's notes about the period which cannot otherwise be obtained. Also print the current view, and a list of the commands.

- v View current unit; display a flash that should be bright enough to catch the eye and make it easier to see where the current unit is.
  - V Version; display the current version and other useless information. Be sure to include the version number when reporting bugs.
  - ^R, ^L Redraw the screen. Redrawing happens automatically most of the time. Keep in mind that *xconq* is a single program, despite opening multiple screens, and attempts to redraw may be ignored for awhile. An argument to redraw will erase the views of mobile units that have not been seen in as many turns as the argument. ^R defaults to erasing units we haven't seen in 10 turns.
  - T, ^T Find a unit. T always prompts for a unit type while ^T will use the current unit type, if any. This will find a unit of the given type, or the next for ^T. An argument to the command gives the number of a particular unit. With no argument, ^T can be used to cycle through all units of one type.
  - # Determine distance; will prompt for another location and will print the distance to that location.
- Additional commands are available for building scenarios, and are described in the customization document.

## PERIOD HELP

The help screens describing unit characteristics include an enormous amount of information. In fact, a period that utilized the full range of capabilities would be too complicated to play, even as a computer game. As a result, the help screen are rather compact and cryptic. For any single unit, there are three tables, summarizing the unit's relationships with resource, terrain, and other unit types. The numbers in parentheses are default values that fill in any blank entries.

### Resources:

- ToBui Amount of resource needed to build the unit.
- Prod Amount produced each turn under best conditions.
- Store Amount that can be carried around.
- Eats Minimum amount consumed during a turn.
- ToMov Amount consumed by moving one hex.
- Hits Amount needed to hit another unit.
- HitBy Amount needed to be hit by another unit.

### Terrain:

- Slowed Move penalty for entering hex with the given terrain type. Default is negative, which prevents movement entirely. 0 means can move in at maximum speed.
- Rand% Chance (in hundredths of a percent) to move randomly in the terrain.
- Hide% Increased difficulty for others to see unit in this terrain.
- Defn% Increased difficulty for others to hit unit in this terrain.
- Prod% Productivity of this terrain for resource production.
- Attr% Chance (in hundredths of a percent) for attrition to occur.
- Acndn% Chance (in hundredths of a percent) for an accident to occur.

### Other units:

- Hit% Chance to hit a unit of that type.
- Damg Number of hit points of damage done when hit successful.
- Cap% Chance to capture unit.
- Guard 1 if capturing unit converted into garrison.
- Pro% Percentage of hit that is prevented from hitting the unit type that occupies, or decrease in chance of hit on unit type transporting this unit.

Holds	Number of units that can be carried.
Enter	Extra moves consumed by entering the transport type.
Leave	Extra moves consumed by leaving the transport type.
Mob%	Transport mobility when carrying unit type.
Bridg	1 if can attack unit type across impassable terrain.
Build	Basic construction time for the unit type.
Fix	Time to repair one hit point of damage to the unit type.

The customization document has additional explanation for some of these (rather obscure) parameters.

## HINTS

*Generally in war the best policy is to take a state intact; to ruin it is inferior to this. -- SUN TZU*

*Attack where he is unprepared; sally out when he does not expect you. -- SUN TZU*

*There has never been a protracted war from which a country has benefited. -- SUN TZU*

The works of Sun Tzu and Clausewitz say nearly all there is to be said on strategy in general. *Xconq* strategy is fairly close to real strategy.

The most important consideration is to conceal your own forces and movements as much as possible. Decoys and feints are worthwhile if they don't draw critical strength away.

Secondly, don't rush to attack with weak forces. Especially over long distances, the defender has the advantage. Wait until you have assembled enough to take and hold a piece of territory, then allow some extra, just in case.

Make a plan, and have some contingency plans ready as well.

Be ready to take advantage of opportunities.

## PERIODS

*Xconq* starts with one period compiled into it. It can also read and interpret other periods. Typically the installer will have built in the period called "standard", for which the description is included below. Other periods include Napoleonic times, Ancient Greece, a somewhat silly futuristic period, an even sillier "Star Wars" period, whose sole reason for existence is to watch death stars blast cities, a "flattop" period featuring carriers, some simulations of board games, and more.

The standard period represents units of about 1945, from infantry to atomic bombs. This is the most familiar, which makes it easier to play, but also more controversial, since historians have many conflicting theories about which kinds of units were most effective. This set has been most influenced by other empire games (thus the greater variety of ships), and the numbers have been honed by extensive playing experience at Utah.

**Infantry.** The infantry division is the slowest of units, but it can go almost anywhere. It is also quick to produce. Infantry is the staple of campaigns - a little boring perhaps, but essential to success.

**Armor.** The armor division is highly mobile and hits hard. Unfortunately, it is limited to operating in open terrain - plains and desert. It also takes longer to produce. Armor can last twice as long in the desert as infantry. Both armor and infantry can assault and capture units; they are the only units that can do so.

**Fighters.** A fighter is a squadron or wing of high-speed armed aircraft. Their fuel supply can be gotten only at units, towns, and bases, so they must continually be taking off and landing. Fighters are not too effective against ground units or ships, but they eat bombers for lunch. Fighters are very good for reconnaissance - important in a game where you can't always see the enemy moving!

**Bomber groups.** Bombers are very powerful, being capable of destroying any unit. Attrition rate in such activities is high, so they're not a shortcut to victory! Bomber performance against other units is less impressive, and of course fighters love to munch on them. Bomber range is greater, but they are slower, taking several turns before they must land. They are also a last-ditch method to escape from a

continent if no seaports are available.

Destroyers. Destroyers are fast small ships for both exploration and anti-submarine activities.

Submarines. The favorite food of submarines is of course merchant shipping and troopships, and they can sink troop transports with one blow. Subs can't be seen by the other side, although their presence might be suspected. Subs are always highly vulnerable to attack by bombers or even fighters.

Troop transports. This is how ground units get across the sea. They can defend themselves against ships and aircraft, but are basically vulnerable. They're not very fast either.

Aircraft carriers. Compensates for the fighter's limited range by providing a portable airport. Carriers themselves are sitting ducks, particularly with respect to aircraft. Fighter patrols are mandatory.

Battleships. The aptly named "Dread Naught" has little to fear from other units of this period. Subs may sink them with enough effort, and a group of bombers and fighters are also deadly, but with eight hit points to start, a battleship can usually survive long enough to escape. Battleships are very effective against units and armies, at least the ones on the coast.

Atomic bombs. The Final Solution; but they are not easy to use. A bomb takes a long time to produce, moves very slowly by itself, is easily destroyed by other units, and reduces the range of bombers that carry it. The plus side is instant destruction for any unit of any size!

Bases. To simplify matters, this can serve as a camp, airbase, and port. Bases cannot build units, although they can repair some damage.

Towns. Towns are the staple of territory. They produce units at the same rate as cities, but are easier to capture.

Cities. Cities are very large, powerful, and well defended. They are basically capital cities, or something in a comparable range. (New York and San Francisco are cities, Salt Lake City and San Antonio are towns.) A city is worth five towns, territory-wise.

Current thinking about optimal strategy for this period differs. In general, blitzkrieg works, and can win the game in a hurry. The problem is to muster enough force before striking. One full troop transport is not enough; the invasion will melt away like ice cream on a hot sidewalk, unless reinforcements (either air or land) show up quickly. Air cover is very important. While building up an invasion force, airborne assaults using bombers and infantry can provide useful diversions, although it can be wasteful of bombers. Human vs human games on the default map generally last about 100 turns, usually not enough time or units to build atomic bombs or battleships, and not a big enough map to really need carriers (although bases for staging are quite useful.)

## CHANGES FROM VERSION 1

Aside from the significant changes (hexes, simultaneity, period descriptions), there are a number of smaller changes that will affect experienced players:

The command to construct a base is now the same as the general build command **'P'** (since bases are units like any other).

The random movement command is gone.

The disband command is now **'D'** instead of **'d'**.

There are probably others I have forgotten about.

## ACKNOWLEDGMENTS

Special thanks must go to Eric Muehle, now at Martin-Marietta, who has been a tireless source of ideas, advice, and playtesting. Mohammad Pourheidari, Bob Kessler, Kevin Deford, Spencer Thomas, Dan Reading, Mark Bradakis, Grant Weiler, Jed Krohnfeldt, Sandra Loosemore, Jimmy Miklavcic, Tim Moore, and others at Utah have also endured initial playtesting, with the apparently endless bugs and misfeatures. Thanks also to Leigh Stoller, who suggested using X, and to Harold Carr, who suggested the postfix language for period descriptions.

Since the first release, many many *xconq* players around the net have sent in literally hundreds of suggestions, fixes, and improvements. Significant contributors include Jim Anderson at CMU; Jay Scott at Swarthmore, who designed the "future" period; John Tonry at MIT, who supplied a great map derived from JPL data; Kurt Hoyt at SEI, who did an X11 port; Julian Onions at Nottingham; Ravi Subrahmanyam at MCNC; and Joel Rives at Georgia Tech, who is working on a large period. Chris Peterson at MIT and Tim Moore at Utah have been essential to the construction and debugging of the X11 interface to version 5. In addition, A.G. Hirai, Jeff Kelley, John Shovic, Dave Pare, Michael Lounsbery, Josh Siegel, Fred Douglass, Cimarron Taylor, and Rick Ledoux have shared a number of good ideas, although not all of them made it into this version.