

```

+-----+
| CCC   000   N   N   QQQ   U   U   EEEEE   SSSS   TTTT |
| C   C   O   O   NN   N   Q   Q   U   U   E       S       T   |
| C       O   O   N   N   N   Q   Q   U   U   EEE       SSS       T   |
| C   C   O   O   N   NN   Q   Q   U   U   E           S       T   |
| CCC   000   N   N   QQ   Q   UUU   EEEEE   SSSS       T   |
+-----+

```

This file contains various information about Conquest, such as a description of the files supplied, and various tips and hints on configuring and playing the game.

NEW USERS: Please read the section *Hints for new players* below for some useful information you should know. Reading the whole document is advised, but this section is probably the most important as far as providing some useful playing tips.

In the sections below, commands you issue to conquest will be enclosed in parentheses '()'. The RETURN key is represented as '\r', and the TAB key is represented as '\t'.

## Conquest Synopsis

Here's an extract from the man page:

### NAME

Conquest - a multi-player real-time screen-oriented space war game

### SYNOPSIS

conquest

### DESCRIPTION

#### i. OBJECT OF THE GAME.

The object of the game is twofold. The short-range goal is to accumulate "kills" by shooting down enemy players. You get one kill point for each enemy ship shot down, plus some extra if the enemy had kills too. The major weapon used to shoot down ships is the photon torpedo.

The long-range goal is to conquer the universe for your team by taking every planet. You take planets by killing off the enemy's armies via bombardment, and then beaming your team's armies down. When all the planets have been taken, the game ends, a new game begins, and the player who actually took the last planet gets his/her name up in lights.

## A Little Background

A quote from Michael Erskine:

We were playing xtrek before there was X... It was known then as Conquest, was written in RATFOR and ran on a VAX...

Personally, I liked it better... but who can account for the tastes of old C programmers?"

Conquest was originally written in Ratfor for the VAX/VMS platform in 1983 by Jef Poskanzer and Craig Leres. I wasted incredible amounts of time playing this game with my friends in the terminal labs at college, and when I actually had a multi-user system running at home (Unixware) I decided to try and translate/port the code to C in Unix. While doing the port, I added several new features including color and Fkey macro support, as well as many other minor 'enhancements'.

For those who played the original Conquest, you may notice that cloaking is more usable, (though not *too* usable), torps have a slightly longer range, the keypad can be used for 1-key steering, the Conquest driver recovers almost immediately from a crash/kill, color and macros are supported, and other small changes. Where possible, I've tried to keep the mechanics and balance of the game identical to the original.

Starting with Version 8.0, Conquest has been split into a true client/server game. Conquest clients connect to a local or remote conquestd server to play.

Starting with Version 8.0.1a, a GLUT based OpenGL client, conquest, is provided.

NOTE: Starting in 12/2017, the curses client has been discontinued and what was formerly "conquestgl" is now just "conquest".

Starting with Version 9.0 (next release as I write this) many of the restrictions in the original Conquest have been removed. In the original Conquest, there were 40 planets and a maximum of 20 ships. The planets were hardcoded into the game, and you had to take all of them in order to conquer the universe.

With v9.0+, almost all of this can be configured by an operator. You can create a Universe with 5 planets or 50, as you see fit. You decide which ones are core planets (needed to take the universe and win the game) and which are home planets (subject to defense by team robots). You can choose names, orbital characteristics and the like as you see fit. The idea was to give as much flexibility in creating a Universe as is reasonably possible.

## General Information

---

### Starting Conquest

---

```
conquest -?
```

You can start the Conquest client with various options. For example type *conquest -?* to see a list of supported options, and what they do.

To connect to a conquest server running on the local host:

```
conquest -s localhost
```

To connect to a conquest server on a remote host:

```
$ conquest -s conquest.radscan.com
```

To connect to [conquest.radscan.com](http://conquest.radscan.com), port 1702.

```
$ conquest -s conquest.radscan.com:1702
```

To contact the default conquest metaserver (default: [conquest.radscan.com](http://conquest.radscan.com)) and select from a list of available online servers:

```
$ conquest
```

NOTE: If you want to connect to a server on your local machine, be sure conquestd is running. See [server.txt](#) for more information on conquestd and running a server.

## Default network ports

---

This is the list of default network ports Conquest uses:

### 1701/tcp - default game server port

conquestd listens on this port for client connections.

### 1701/udp - default game server port for UDP location data

conquestd listens on this port for client connections to determine if the client and server can both do UDP. The server will also send UDP motion data to the client(s) from this port.

UDP is only used for ship motion data from the server to the client. The client only sends data to the server on this port during UDP negotiation.

If you are running a server, you can use the '-p' option to conquestd to have the game run on a different port other than the default of 1701.

### 1700/udp - metaserver update port

The conquestd server sends updates to conqmetad on this port (when run with '-m')

### 1700/tcp - metaserver query port

When you (or a conquest client) connects to this port on the metaserver host (default: [conquest.radscan.com](http://conquest.radscan.com)), the metaserver dumps the current server list out on this port and disconnects.

## Server game flags

---

The server operator can set various game flags via the conqoper Options menu. These flags are displayed on the conquest login screen (when connecting to a server), as well as when the user selects the Options menu from from within the game (while already connected to a server).

Currently the following flags are available.

### Refit

Refits are allowed. You can refit to one of the 3 types of conquest ships when you have at least one kill, and are orbiting a team owned planet.

### Vacant

Vacant ships are allowed. A ship goes vacant when a client exits the game without self destructing first, while flying a ship.

When vacant ships are disabled by the oper, such ships are immediately killed by a 'lightning bolt'.

### SlingShot

The SlingShot bug is enabled. This is a towing bug that existed in the original VMS/RATFOR version of conquest. It's a fun bug, and I received several emails about it when I inadvertently fixed it in a later version

without understanding it's true significance.

If you don't know what it does, well... figure it out! You'll need a buddy to make it work though ;-)

## NoDoomsday

When set, the doomsday machine is prevented from randomly starting up and trashing the universe.

## Killbots

When this option is enabled, robots are created with a random kill point value. The kill points a ship has affects the efficiency of it's engines, the strength of it's weapons, etc.

Normally when a robot is created (by attacking an unguarded homeplanet, etc), it is initialized to 0 kills. When this option is enabled, a kills value is randomly selected. This means that a new robot might prove to be much more deadly than the default 0 kill robot.

This may not be a good option to enable on a public server where newcomers play. Fighting a 25 kill robot takes some skill ;-)

## SwitchTeam

When enabled, users can switch teams at the conquest main menu.

When not enabled, a random team (Federation, Klingon, Romulan, Orion) is chosen for you.

## NoTeamWar

When enabled, users are not allowed to declare war on their own team.

## NoDrift

When enabled, the Drift bug is disabled. Like the Slingshot bug, if you don't know what this is, figure it out :)

## Closed

When enabled, the game is marked as closed. This means new users cannot login. Existing users can login and play only if the PLAYWHENCLOSED flag has been enabled in their user record by conquer.

If there are already people logged in and playing on a server when the server is marked as closed, nothing will happen to them – it will not kick out users that are already in the game.

## Meta Server

---

With Conquest v8.0 and better, a meta server is supported. There is one running at [conquest.radscan.com](http://conquest.radscan.com).

If you want to see what servers are available, run conquest without any options.

This will query the meta server and list the currently active servers. From here, you can select one and connect.

There is a cgi-bin perl script on the Conquest website where you can also get this info:

[Server List](#)

If you want to run a server that other Internet players can play on, and you want them to know about it, you will want to run your conquestd server with the '-m' flag.

```
$ conquestd -d -m
```

Depending on your network topology, you may also need the '-N' flag if the Internet has a different idea about your server hostname than you do internally. For example, here, [conquest.radscan.com](http://conquest.radscan.com) is an external only address, so I must explicitly specify it to the metaserver, or it will use the address the update packet came in on (which no-one would be able to resolve and connect to). For example, here is the complete line I use on my server to start conquestd:

```
conquestd -d -u nobody -m -N conquest.radscan.com
```

Please make sure your server is actually reachable from the Internet before advertising your game to the meta server.

## **/opt/etc/conquest/conquest.conf, ~/.conquest/conquest.conf**

---

After running the Conquest client for the first time, a file called ~/.conquest/conquest.conf should have been created. Look at this file (it's self-explanatory) for various options you can set.

Conquest Operators (or CO's) should look at /opt/conquest/etc/conquest.conf for a few system wide options you may or may not want.

As of version 7.0, all options (user and system-wide) can be viewed and edited using the (O)ptions Menu. This menu is available in the conquest/conqoper main menus, as well as being accessible from within the game.

## **Recording Games**

---

With Conquest 8.0, recording has been significantly re-worked. '.cqr' files created prior to v8.0 are not compatible with v8.0+, sorry.

The new format uses the same packet protocol that the clients use. This means recording files (cqr's) are *much* smaller and much more efficient (cpu-wise) during the recording process, than they were with v7.x.

Another benefit is endian safety. With the old format, recordings could only be played back on the same machine architecture the cqr was created on. Since the recording data is now based on the packet protocol, recordings are now completely endian safe (network byte order is used). So now, some poor slob on his Sun SPARC can play back a recording I make on a intel machine and vice-versa.

Recordings can only be made by the server. If you wish to record a game client-side, I recommend screen capture/recording software like kazam.

Server recordings are created by conquestd on the server machine whenever a CO with the OOPTION\_OPER set in their user record, sends a command to GOD from within the game containing:

```
/recon
```

Server recording files will be saved in:

```
INSTALL_PREFIX/var/conquest-rec-.cqr
```

To turn off server recording, send the following to GOD from within the game:

```
/recoff
```

## Server recordings

Server recordings are complete (like the old recordings made in previous versions of conquest), since the server has full access to the common block.

All active ship/planet/etc data is stored in server recordings. You can watch any ship during playack with a server recording and get all info on the ship (heading, fuel, temps, etc).

To replay a recording:

```
$ conquest -P somefile.cqr
```

If libz and it's development header files are available on your system when building conquest, game data will be recorded using gzip compression. It's a good idea to use this. Replay can read either uncompressed or gzip compressed files, and can be played back using the client.

If your client was not compiled with libz support, then you will need to uncompress any compressed .cqr files you may have before you can replay them. A simple way is:

```
$ gunzip -f <somefile.cqr >somefile-nocompress.cqr
$ conquest -P somefile-nocompress.cqr
```

## Hints for new players

Here are some general hints for new players. There's more to playing than what is described in this section, but it should be a good start.

### Moving around (Navigation)

With the version 8.1.2f or better OpenGL client, you can click the middle mouse button (by default) in the viewer to set a course in that direction.

For the keyboard, using the direction keys ('qwleadzxc') to the (k)ourse or weapons commands can be faster than specifying the direction in degrees:

```
Q W E
 \ | /
A--+-D
 / | \
Z X C
```

These direction keys can be used to set a course and aim your weapons.

You can use them singly, e.g. 'd' would be 0 degrees, 'q' would be 135, etc. You can also use them in combination: "ed" would be halfway between 'e' and 'd', which is 22.5 degrees; "edd" is like 'e' + 'd' + 'd' / 3

You can set course, lock onto, and automatically enter orbit around a planet by typing the planet name (or first 3 unique characters) followed by [TAB] as input to the (k)ourse command. You will automatically enter orbit when you get close enough to your destination.

The Keypad keys/Arrow keys can be used for 1-key steering, which can be faster in battle. A mouse can also be used (default middle mouse button) by clicking the mouse in the viewer.

Of course, once you have set a direction, you need to set a warp speed if one is not already set, or you are not

going to actually go anywhere. Use the number keys (0) to (9) to set the warp. Use (=) to set the maximum warp your ship can currently go.

## Orbiting a planet

The easiest way to get into orbit around a planet is to lock onto it, set a warp, and let the ship automatically enter orbit when you are close enough, as mentioned above.

For example, to 'lock' onto Janus and enter orbit, enter (kjan\t). When you are close enough, your ship will decelerate and automatically enter orbit.

To enter orbit manually, get close to the planet, slow down to warp 2 (at the fastest) and hit (o). You cannot enter orbit if you are going faster than warp 2.

See the section *Moving around (Navigation)* for more information on navigating around.

## (i) Info command

You can use the (i) Info command to query things like a ship or a planet's status, bearing and distance, as well as a lot of other information about the universe. Here's a list:

Command	What
ne	nearest enemy
ns	nearest ship
nts	nearest (friendly) team ship
np	nearest planet
nep	nearest enemy planet
nrp	nearest friendly (repair) planet
nfp	nearest friendly class-M (fuel) planet
nap	nearest planet with available armies (greater than 3 total armies)
ntp	nearest planet owned by your team
wp	weakest planet not owned by your team
hp	home planet for your team
sN or just N	ship N (where N is a integer number)
planet name (or first unique characters)	planet information

All of these can be abbreviated to their shortest unique string. Also, for the planets, you can type a number after the special string to specify an army threshold; that is, planets with less than that number of armies won't be considered. For example, *na8* specifies the nearest planet with 8 (eight) or more armies, *nf14* is the nearest fuel planet with 14 or more armies, *nep1000* is the nearest non-scanned planet.

## Shields

Shields are important for protecting your ship. Once your shields are down, then your ship's hull will start taking damage.

When your shields are up, they consume more power and cause your engines to heat faster.

For this reason, *only* keep your shields up when you are in danger - ie: an enemy ship is nearby, or you are close to an enemy planet, torps, or a Sun, etc.

Don't run around the universe with your shields up all the time. You are just wasting fuel and heat.

On the other hand, when an enemy gets close, don't forget to raise them :)

Your shields will repair at twice the normal rate when lowered.

## Energy Allocation

In Conquest, you can assign an energy allocation that is split between your engines and your weapons. The max allocation percentage is 30/70.

When you are just flying around, set your allocation so your engines get most of the power. This can be done with (A30\r). This way your engines will be as efficient as possible - using less fuel, generating less heat, and making your ship more maneuverable at high speeds.

When bombing, do the opposite. Having your weapons have the maximum power allocation increases your army kill rate. This can be done with (A70\r).

When dogfighting, you will probably want to quickly switch between the two. I would use a Macro for this (see the *Macro Keys* section). In this case, what you want to do is keep maximum power for your engines, but switch to max weapons whenever you fire them. This increases their power, and potential damage to an enemy.

I personally use a Macro for most phaser firing and torping (except for aiming them). This macro switches to max weapon allocation, fires the weapon, then switches back to max engine allocation.

Most efficient. Many times when you are dogfighting someone else (or a robot) the first one to overload or run out of fuel loses. It's important that you are not the one that this happens too :)

Some battles I have fought have depended entirely on the efficient use of fuel and heat resources. A good player will always try to get you to waste fuel and heat up. Robots, being fairly stupid, can be goaded into wasting fuel and heat too, if you are careful.

## Bombing

Bombing is the usual way to get kills (unless you are good with robots).

To bomb a planet, raise shields and go into orbit. Set weapons allocation to max (A70\r), and use the (B) command.

If you want to bomb a team owned planet (those that are not 'self-ruled'), you will need to declare war with the team first (W). Bombing a self-ruled planet automatically makes you at war with that planet.

The more armies a planet has, the faster it will damage you. Keep this in mind.

When bombing, set your weapons allocation to max (A70\r). This will increase your army kills per bomb run.



When you have bombed the planet down to 3 armies, you now have to **take** it. To take a planet, you must go and get some of your team's armies from one of your team's planets and beam them down. You will need at least 4. 3 for killing the remaining 3 enemy armies, and 1 to claim the planet as your own.

While bombing, break off and repair when your damage gets around 50-70%. As you get more experienced, you can push the line a bit. But it's important to know that the more damage you have, the slower you will be able to run. With 70% damage, your maximum speed will be about warp 3. On a 100 army planet, this may not be fast enough for you to move away from the planet before the armies overwhelm your shields and you explode.

Exercise caution :)

Bombing a home system planet of a team will create a robot defender. Be ready for this. If you accidentally end up creating a robot that you do not/can not fight, just cloak, raise your shields and go to warp 0. If the robot wasn't too close to you when you did this, you should survive and the robot will lose interest :)

By default, home systems, and their planets are the following:

Team	Home Planets
Federation	Earth, Telos, Omega
Klingon	Klingus, Leudus, Tarsus
Romulan	Romulus, Remus, Rho
Orion	Orion, Oberon, Umbriel

All other planets are self-ruled when a game is initialized, and will not create robot defenders when bombed (even if owned by another, non self-ruled team). Newcomers should probably avoid bombing home system planets until they are ready to handle robot defenders.

By default, all planets, with the exception of Altair, Hell, and Jinx, must be taken in order to conquer the universe.

NOTE: With the latest version of Conquest (tentatively called V9.0), all of this is configurable by the Server Operator. The Operator can define a Universe where any planet is a Home planet or Core Planet.

Home Planets are defendable by a Team robot. Core planets are planets that must be taken in order to win the game (Conquer the Universe).

In the client, the (/)Planet List command will list each planet.

Planets preceded by a + sign indicate a Core planet. Planets with a team character ('F', 'K', 'O', 'R') preceding the name are Team Home Planets.

## Repairing

When you are damaged, you will want to repair as soon as you are able, so that you can regain your max warp and efficiency.

When not under attack, you will repair at a 'nominal' rate - which will be quite slow.

Use the (R) command to go into repair mode. This sets you at warp 0 and lowers your shields - so don't do it in the middle of battle.

Repairing is faster this way.

If there is a friendly planet nearby, go into orbit around it before using (R). You will repair even faster.

You will also cool down much faster when in orbit about a friendly planet.

## Fuel

Fuel is important. Make sure you don't run out of it. When sitting in space, and not moving, your fuel will regenerate, though somewhat slowly. To refuel faster, go into orbit around a friendly Class M planet.

## Cloaking

You can cloak your ship if you wish with the (C\t) command.

Cloaking consumes a lot of power and heat. If you are moving, you will exhaust your fuel or overload your engines fairly quickly. How fast this happens depends on your ship type, and how fast you are moving.

If you put your shields up, you will overload even faster. Overloading your engines is really bad in battle for obvious reasons.

If you are moving when you are cloaked, you are detectable - but only to an approximate location.

If you are at warp 0, and you cloak, you cannot be detected at all. This is a good trick to use when you want to ambush an enemy who is approaching (assuming he does not already know you are in the vicinity).

This is also a good way to escape from robots if you find you are in trouble. You can cloak and go to warp 0. Then even robots cannot detect you. This may give you the time you need to heal a little, and perhaps the robot will wander off somewhere and give you some breathing room.

Keep in mind that depending on ship type and your kills score, decelerating can take a little time. So, if you are trying to evade a robot, and he is very close, don't go to warp 0 and cloak. While you are decelerating you are still detectable enough for the robot to steam-roll right over you. Get a little distance from it first.

## Carrying armies

You should only carry armies when you are going to take over a planet. If there are a lot of enemy armies on the planet, bomb them down some before bringing armies.

Carrying armies will increase fuel consumption and heating, so only carry them when you are going to use them fairly quickly.

It's a real bummer to have to get into a dogfight when you are carrying armies. Unless you are good and your opponent is bad, you will probably not survive since you will most likely run out of fuel or overload before he does.

## Detonating enemy torps

With the (d) command, you can try to detonate enemy torps before they hit you. This command consumes fuel and heat, and will only work on torps that are relatively close (<1000 CU's) to you.

It is better to detonate torps before they hit you, to limit damage. If it is possible for you to simply move out of the way of oncoming torps, then doing so is more efficient than detting them.

If you can move out of the way rather than det them, the enemy ship will have wasted fuel and heat firing them, and will then also have to det them himself in order to free up torps slots to fire more - while you have expended far less fuel and heat to evade them.

Getting the enemy to waste fuel and heat faster than you is an important strategy in battle.

## Using torpedos

Torpedos are probably the most used weapon in Conquest. When fired, they head in the direction they were fired until they hit an enemy ship, or timeout (about 50 seconds).

The damage they inflict depends on the ship type that fired them, the number of kills the ship has (more is better), as well as the weapons allocation in effect when they were fired.

For the maximum damage potential, you should set weapons allocation to 70% (A70\r) when firing. Maybe use a macro. :)

With the 8.1.2f or newer client, you can click the right mouse button (by default) in the viewer to fire a torpedo in that direction.

## Detonating your own torps

You can only have 9 torps out at a time. If you have fired them all, and they missed, you should self-detonate them (D) so that you can free up slots to fire more. Hopefully with better aim this time. :)

## Using phasers

Phasers are a close-in weapon. If the ship you are firing at is not in RED ALERT range (about 1000 CU's) they won't have much (if any) effect, and you will just waste fuel and heat using them.

They are *very* effective close up though, so don't abandon them in favor of using torps only.

With the 8.1.2f client or later, you can click the left mouse button (by default) in the viewer to fire the phasers in that direction.

You can only fire a phaser about once per second. When in really close combat, alternate between firing your phasers and torpedoes.

## Kill points

In short, the more kill points your ship has, the better.

You get kills by bombing planets, and blowing up enemy ships.

The number of kills you have will affect your ship in the following ways:

- The more kills you have, the more efficient your weapons and engines will be, in terms of fuel consumption and heating.
- the more damage your weapons can inflict.
- acceleration and deceleration will be faster. You will also be able to turn quicker at higher speeds.
- See also, the *Ship Strength* section for more information.

If you are fighting enemy players, you will probably want to concentrate on those with the highest number of kills. If you let someone get a huge number of kills, it will just get harder to destroy them unless they make a mistake, or you manage to ambush them.

## Robot scanning range

Robots have a somewhat limited scanning range. Basically, if you are within around 6500 of an enemy robot, he can scan you and will come after you if he can.

If you are bombing planets and there are other enemy robots around, you might want to do frequent info commands on them (ine\r) to make sure you stay outside this rough radius. Otherwise, they will see you and come after you.

This is especially good advice if you are not good at fighting robots :) Avoid them if you can.

## Planets

---

You need at least one kill before you will be able to transport armies to, or from, a planet.

Core planets are those that need to be conquered in order to take the universe. The (?)planet list option in conqoper/conquest will identify core planets with a '+' sign. Currently, there are only 3 non-core planets, that while not necessary to conquer the universe, can provide some strategic advantage. These are Altair, Jinx, and Hell. Particularly Altair. Keep an eye on Altair ;-)

Note, with the latest version of Conquest (tentatively called V9.0) all of this is configurable by the server operator.

Use them together. Use them in peace.

## Suns

---

Suns are hot. Don't fly through them unless it absolutely, positively, has to be there overnight. Hint: Robots don't seem to worry about suns. This can be used against them.

## Cloaking

---

Cloaking can be very useful in battle. Unfortunately, it's expensive in terms of fuel and engine heating. But at warp 0 however, it can be very nice.

## Tractor beams

---

One use I've seen for them so far is to drag a helpless ship into the sun, so it's death can be as humiliating as possible. There *are* a few other interesting uses for them as well ;-)

If the server has the SlingShot flag enabled, even better ;-)

As of V9.0, *tow-chains* are possible.

## Ship Strength

---

Different teams have different strengths. Romulans have the best weapons, but the worst engines. Orions have the best engines and the worst weapons. The Federation and Klingons are in-between. In original Conquest, this is what you were stuck with.

But now, if the server has the Refit flag set, you can change your ship type, regardless of which team you are on, provided you are orbiting a team owned planet and have at least one kill.

The number of kills your ship has will determine how much more efficient your weapons and engines are than the base efficiency you started out with.

When your kills count reaches 50, your efficiency will be double what you started out with. It's very nice to be in a ship with a lot of kills. The problem is, at least in our games, we tend to attack whoever has the most kills - it's

a wise move for self preservation if the ship in question isn't on your team ;-)

Even if you have high kills, it's difficult to survive against the continuous onslaught of a couple of determined foes, unless you run and cloak alot.

## Refitting

---

If the server you are connecting to has enabled the Refit flag, you will be able to refit your ship to a new ship type.

Basically, when you enter the game, and join a team, your ship is the same type of ship (in terms of weapons and engine performance) that is the default for your team - like the original Conquest.

With Refit capability, once you have at least one kill, and are orbiting a team owned planet, you can use the (r) Refit option to select a different ship type. The current ship types, and their equivalence with the traditional types assigned to a team are listed below:

Default Ship Type	Traditional ship type assigned to team
Scout	Orion - strong engines, weak weapons
Destroyer	Federation/Klingon - good engines, good weapons
Cruiser	Romulan - weak engines, strong weapons

The type of a given ship can be determined by doing an (i)nfo on it, or by using '/' Player List to look at the list of currently active ships, and note the character ('S', 'D', 'C') following their ship number.

The ability to refit is controlled by the server operator with the 'Refit' option in the conqoper options menu.

## Leaving the game

---

To exit Conquest, normally you must self-destruct, or be killed.

In a hurry, you can also exit Conquest quickly with the SIGQUIT signal (usually the Control-\ key).

If the server operator has enabled the 'Vacant' flag in the conqoper Options menu, then your ship will be left intact on the server, so you can reconnect to it in the future.

If you exit this way, I'd be careful where you leave your ship.

If the 'Vacant' flag has not been enabled by the server operator, then your ship will be immediately killed by a lightning bolt on the server.

## Teams

---

When you first enter Conquest, it will randomly select a team for you, before bringing you to the main menu. At this point, you can switch teams with the (s)witch teams option if you wish (provided the server has the 'SwitchTeams' flag enabled).

Remember, different teams have different strengths and weaknesses. Federation and Klingon teams are pretty middle-of-the-road as far as engine/weapons efficiency goes, while Orions have better engines, and Romulans have stronger weapons. But you can always Refit if the server allows it...

## Combat

---

Taking on a robot is quite a bit different from taking on a human player. With a robot, in time you learn it's strategy, and compensate for it. After you've done it a few hundred times, robots aren't too much of a challenge, if you don't do anything brave and stupid :-).

People on the other hand, tend to adapt to your strategies, forcing you to come up with new ones.

There are various strategies that can be employed effectively against your opponent. A common one we used to use, is the 'lame-duck maneuver'. If you take alot of damage, though you have plenty of fuel and your weapons are cool, sometimes you can trick an enemy into thinking your really hurting – by limping away at warp 2 with your shields down for example.

He'll do an info on you and see your damaged, or he might think you're out of fuel, and therefore, an easy kill. Sometimes you can surprise him ;-) It's simple, but often effective with a player determined to 'finish you off' carelessly.

Cloaking can be used to excellent effect on an unsuspecting opponent. I leave it up to you to explore the possibilities.

Getting your opponent to waste fuel and heat is also a good idea if you can arrange it.

## Macro Keys / Mouse Macros

---

Macros are sequences of Conquest commands that are issued when a Function Key (Fkey) is hit. On PC hardware, these are the F1-F12, SHIFT F1-F12, and CTRL F1-F12 keys.

With the GL client, version 8.1.2f or better, support for assigning macros to mouse buttons is also provided. Modifiers like Alt, Control, and Shift can also be used with the mouse buttons. Up to 32 buttons are supported with any combination of the 3 modifiers (or no modifiers) giving you a maximum of 256 assignable mouse macros. If you have a 32 button mouse that is :)

Mouse and Key macros are defined in your ~/conquest/conquest.conf file. Users can edit their macro keys from within Conquest using the (O)ptions Menu.

Here is an example for the F1 function key as it would appear in the ~/.conquest/conquest.conf file:

```
macro_f1=dP\r
```

Which makes my F1 key detonate enemy torps (d), and fire a spread of 3 torps in the last direction I fired (P\r).

The mouse only works when playing the game (in the Cockpit) and the cursor is within the viewer window.

Three default mouse macros are provided to give you a taste, and will be saved in your conquest.conf file the first time you run an 8.1.2f or better version of the client. They are:

- mouse button 0 (left): Fire phaser at *angle*
- mouse button 1 (middle): Set course to *angle*
- mouse button 2 (right): Fire Torp at *angle*

Of course you can redefine these, as well as add others. With mouse macros, a special character sequence, \a can be used to represent the angle of the cursor relative to the center of the viewer when the button was pressed.

See the mouse macro comment block in your conquest.conf file for a description of the format.

There are many other interesting and useful combinations that I won't detail... After all, choosing the right

macros and using them well is an important part of the strategy you employ against your opponents.

## The Robots

---

### Conqstrat

The AI code used by the robot ships is the original strategy table that was generated by the conqstrat.r program with the exception of one new rule that lessens robot sun-deaths somewhat.

The conqstrat program can be used to modify the Robot strategy tables if you don't like the supplied rules. You can have a maximum of 32 rules. Conquest is supplied with a file called conqrule that describes in a simplistic 'language' how a robot should behave under certain conditions... You can edit this file, and use conqstrat to generate a new conqdata.h file.

The following command will generate a new conqdata.h file. You can then recompile Conquest to get the new default strategy table. After compiling and installing, remember to (l)nititalize the (r)obots to update the common block copy of the strategy table with the compiled in version.

```
conqstrat -o conqdata.h <conqrule
make all
make install
```

Of course, this is only of use to server operators.

### Combat

To new users, the robots may seem tough. Gleeefully harsh, even. But they are predictable, and can be taken once you learn their 'strategy'. One thing to remember, NEVER attack one head-on (let alone 2 or three...) unless you are experienced. They are much faster on the trigger than you are, and you'll probably lose.

They particularly like to phaser the crap out of you when in range.

### Creating Robots

Robots are created one of two ways:

- You attack a home-system planet of an opposing team, and there are no team players around to defend it. Presto, one pissed robot headed your way.
- A Conquest Operator fires up conqoper and creates some with the (r)obot menu option. A user marked as a Conquest Operator can also create robots by sending special messages to GOD from within the game.

## The Doomsday Machine

---

The Doomsday machine (if activated) *can* be killed. It probably requires that you've seen the (old) Star Trek episode called 'The Doomsday Machine' though.

It can also be rather annoying after a time or two though. If you get tired of it, you can set the 'NoDoomsday' flag in the conqoper Options menu to true, which will prevent it from randomly starting up and wasting the universe. I'd recommend this for public access/beginner type servers.

## Using conqoper

---

### The semaphore status line

The semaphore status line (line 2) in conqoper can give you useful information on the locks used by Conquest to prevent simultaneous writes to the common block. The following is an example line, labeled by the numbers 1-9 above it.

```
1      2  3      4 5      6  7      8 9
-      -  -      -  -      -  -      -  -
MsgCnt = 268(25116:0) CmnCnt = 4693(25116:0) Last: Nov 28 13:10:38
```

KEY:

- i. status for the messaging semaphore. preceded by '\*' if currently locked
- ii. number of semops on this semaphore
- iii. PID of last process to alter this semaphore
- iv. number of processes waiting for the semaphore to become zero. ie. the number of processes waiting to acquire a lock. This should be 0 99.9999% of the time.
- v. status for the common block semaphore (everything except messages).
- vi. number of semops on this semaphore
- vii. PID of last process to alter this semaphore
- viii. number of processes waiting for the semaphore to become zero. ie. the number of processes waiting to acquire a lock. This should be 0 99.9999% of the time.
- ix. time and date of last semop.

## Leaving a screen in conqoper

When watching another ship in conqoper or editing a player or planet, use 'q' to quit.

In most other screens, you can use RETURN or SPACE to quit. I know... some consistency is needed.

## The installed binaries

Here are the files installed by Conquest, and a brief synopsis of what each does. All paths are relative to the installation directory (/opt by default).

### bin/conquest

This is the freeglut/OpenGL based conquest client. Type 'conquest -?' to see a list of options.

The client can also be used to replay conquest recording (.cqr) files using the '-P' option to conquest.

NOTE: As of 12/2017, the curses client (originally called "conquest") has been discontinued and conquestgl has been renamed to conquest.

### bin/conquestd

This is the conquest server. Type 'conquestd -?' to see a list of options. See [server.txt](#) for information on the server and how to set one up. If all you want to do is use the conquest client to fight on remote servers, you do not need to run your own server.



In addition, if run with the '-m' flag, conquestd will notify the metaserver at [conquest.radscan.com](http://conquest.radscan.com) of your game, so that others that query the metaserver will see your server and might jump in.

Oh, and if your conquestd server is not accessible from the Internet, please don't register it with the metaserver.

## **bin/conqmetad**

The Conquest metaserver. You should never need to run this unless you have a group of conquest servers on your private network that you do not want to advertise to the world. The definitive metaserver for public Internet conquest games is running at [conquest.radscan.com](http://conquest.radscan.com).

## **bin/conqdriv**

The universe driver process. A driver is kicked off whenever someone enters Conquest and a driver isn't already running. A normal user cannot start the driver manually. (You should never have to.) An operator can manually start the driver for debugging purposes by supplying the '-f' option.

## **bin/conqoper**

This is the Conquest Operator (CO) curses-based program that allows suitably privileged individuals to control, monitor, and modify the behavior of the game. The root user is always a CO and therefore can run the conqoper program. If you want to allow other people to be able to run conqoper, you will need to add them to the conquest group in `/etc/group`.

If you want to allow specific remote players to be able to issue certain operator commands from within the game, (e)dit their usernames in conqoper and set the 'Conquest Operator' flag.

Be careful who you give CO status to, a bad CO can cheat, or otherwise disrupt a game. In addition, due to the fact that a CO is a member of the conquest group, a bad CO will be able to trash the common block, as well as other undesirable things.

A CO with permission to overwrite the system-wide `conquest.conf` file can call conqoper with the '-C' option to update the file with a newer version. This is done by default when building the source and doing a *make install*.

User-level `~/conquest/conquest.conf` files are always updated automatically when conquest is run.

## **bin/conqai**

This program allows a CO to take over robot control from the Conquest driver for debugging purposes. Don't run it if you don't know what it does. It will disable robot AI control by the driver. After running conqai for some purpose, be sure to re-run it with the '-r' option to return control of the robots to the Conquest driver when you're done.

## **bin/conqinit**

This is a utility program that can be used to parse the various CQI configuration files (like `conqinitrc`) in conquest and show any parse errors. It's mainly used for debugging and development.